

VisualAge Pacbase



Installation Guide Windows Server & Client Components

Version 3.5



VisualAge Pacbase



Installation Guide Windows Server & Client Components

Version 3.5

Note

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

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

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

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

Seventh Edition (March 2014)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.5

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

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

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

© Copyright IBM Corporation 1983,2014.

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

Contents

Notices	vii	Chapter 4. Installation/Reinstallation of Client Components	23
Trademarks	ix	Things to Know Before Installing	23
Chapter 1. Foreword	1	Root Directory	23
Introduction	1	Installation Startup.	24
VisualAge Pacbase Architecture	1	Fundamentals of VA Pac Client-Server Communication	24
Contents of Supply	2	Administrator & Developer workbench	26
Bibliography	3	Open Jade and Tidy / Publishing facility	29
Chapter 2. Prerequisites	5	eBusiness Tools	30
Prerequisites for the Server Environment	5	VisualAge Pacbase WorkStation	31
Hardware and Software	5	Web Application Models (WAM)	37
Disk Space	5	Middleware	37
Prerequisites for the Client Environment	6	Editing Communication Parameters	39
Hardware	6	The bases.ini File	39
Disk Space	6	The vaplocat.ini File	42
Software	6	Component Updating, Modification, or Removing.	46
Communications	7	Chapter 5. Tests	47
Chapter 3. Installation of the Server Components	9	List of Utilities	47
Installation	9	Installation Tests	47
System Installation.	10	Generation-Print, Online and Batch Update Tests	48
An element of the System: the Procedure Definition	11	Administration Database Procedures Tests	48
Execution mode.	12	Development Database Procedures Tests	48
Repository Installation	14	Extraction-Utility Tests	49
Administration Database.	14	Chapter 6. Server Re-installation	51
Development Databases	15	Reinstallation	51
Access key entering	16	Uninstall the system	51
Backup of the Administration Database	16	Data deletion	51
Create New Database	16	Chapter 7. Retrieval - Exchanges between 2.n & 3.n Databases	53
Configuration parameters	16	Retrieval of VisualAge Pacbase 2.0 and 2.5	53
Unreference a Database	17	Foreword	53
Components dates list.	17	Notes	53
INSL - Execution Script	17	Operations to be Performed.	54
Connection	19	Retrieval of User Parameters (PE25)	56
Starting and Stopping a Server.	19	PE25 - Introduction	56
Database under DSMS control	20	PE25 - Input / Processing / Results	57
Connecting to a 3270 Emulator.	21	PE25 - Description of Steps	58
Monitor for the Management of Listeners	21	PE25 - Execution Script	60

Retrieval of the Development Database (PC25)	62	MB30 - Introduction	106
PC25 - Introduction	62	MB30 - Description of Steps	107
PC25 - Notes on Data Retrieval	62	MB30 - Execution Script	107
PC25 - Input / Processing / Results	64	GY Transactions exchanges between 3.n & 2.n (GY30)	108
PC25 - Description of Steps	64	GY30 - Introduction	108
PC25 - Execution Script	68	GY30 - Description of Steps	109
Generation-Print Commands Retrieval (PG20)	71	GY30 - Execution Script	109
PG20 - Introduction	71	Procedures - Summary Table of Changes	110
PG20 - Input / Processing / Results	72	Retrieval of VisualAge Pacbase 3.0	113
PG20 - Description of Steps	73	Operations to be Performed	113
PG20 - Execution Script	76		
Generation-Print Commands Retrieval (PG25)	80	Chapter 8. RPP Utilities	115
PG25 - Introduction	80	MIAM - Migration Help Function	115
PG25 - Input / Processing / Results	81	MIAM - Introduction.	115
PG25 - Description of Steps	81	MIAM - User Input	124
PG25 - Execution Script	85	MIAM - Description of Steps	126
Retrieval of PJ Transactions (PJ25).	88	MIAM - Execution Script	126
PJ25 - Introduction	88	MIBA - Batch Migration.	127
PJ25 - Description of Steps	89	MIBA - Introduction	127
PJ25 - Execution Script	90	MIBA - User Input	127
Retrieval of PP file (PP25)	91	MIBA - Description of Steps	130
PP25 - Introduction	91	MIBA - Execution Script	135
PP25 - Input / Processing / Results	91	MIBR - Batch Migration of the Context	138
PP25 - Description of Steps	92	MIBR - Introduction	138
PP25 - Execution Script	94	MIBR - User Input	138
Retrieval of passwords (UTMP)	96	MIBR - Split of Libraries	142
UTMP - Introduction	96	MIBR - Description of Steps	143
UTMP - Description of Steps	96	MIBR - Execution Script	147
UTMP - Execution Script.	96	MIBJ - Migration of the journal	150
Retrieval of Pac/Transfer Parameters (UV25)	97	MIBJ - Introduction	150
UV25 - Introduction	97	MIBJ - User input.	151
UV25 - Input / Processing / Results	97	MIBJ - Description of steps.	153
UV25 - Description of Steps.	98	MIBJ - Execution Script	158
UV25 - Execution Script.	100	MIMA - Migration of the Macros	161
MB Transactions exchanges between 2.n & 3.n (MB25)	102	MIMA - Introduction.	161
MB25 - Introduction	102	MIMA - User Input	161
MB25 - Description of Steps	102	MIMA - Description of Steps	162
MB25 - Execution Script	103	MIMA - Execution Script	164
GY Transactions exchanges between 2.n & 3.n (GY25)	104	MITH - Migration of the Thesaurus	166
GY25 - Introduction	104	MITH - Introduction	166
GY25 - Description of Steps	105	MITH - User Input	166
GY25 - Execution Script.	105	MITH - Description of Steps	167
MB Transactions exchanges between 3.n & 2.n (MB30)	106	MITH - Execution Script	168
		MIUS - Migration of the Users	170
		MIUS - Introduction	170
		MIUS - User Input	170
		MIUS - Description of Steps	171
		MIUS - Execution Script	173

MILA - Migration of Administration Error		Development Database Backup Files	213
Messages	175	Modules - Specific Files	213
MILA - Introduction	175	Pac/Impact	213
MILA - User Input	175	DSMS.	214
MILA - Description of Steps	176	PAF	214
MILA - Execution Script	178	Complementary Libraries and Files	214
MERT - Transformation of Model entities	179		
MERT - Introduction.	179	Chapter 10. Appendix	217
MERT - User input	180	Installation of the Administration Database	
MERT - Description of steps	180	Model	217
MERT - Execution JCL	182	VINS - Introduction	217
MIAx - Optional control cards	183	VINS - Input / Processing / Results	217
MIA1 - Program Generation	184	VINS - Description of Steps	218
MIA1 - Introduction	184	VINS - Execution Script.	220
MIA1 - User Input	184	Installation of the Development Database	
MIA1 - Description of Steps	184	Model	222
MIA1 - Execution Script	187	VINS - Introduction	222
MIA2 - Screen Generation	189	VINS - Input / Processing / Results	222
MIA2 - Introduction	189	VINS - Description of Steps	223
MIA2 - User Input	189	VINS - Execution Script.	225
MIA2 - Description of Steps	189	GS/HE/ZS Files Initialization Utilities	227
MIA2 - Execution Script	191	LDGS - Introduction	227
MIA3 - Client Generation	193	LDGS - Description of Steps	227
MIA3 - Introduction	193	LDGS - Execution Script	227
MIA3 - User Input	194	LDHE - Introduction.	228
MIA3 - Description of Steps	194	LDHE - Description of Steps	228
MIA3 - Execution Script	196	LDHE - Execution Script	229
MIA4 - Server Generation	198	LDZS - Introduction	229
MIA4 - Introduction	198	LDZS - Description of Steps	230
MIA4 - User Input	198	LDZS - Execution Script	230
MIA4 - Description of Steps	199	Retrieval utilities	231
MIA4 - Execution Script	201	UTU1 UTU2 - Adaptation of 'UNS'	
TRDQ - Conversion of DBD Blocks into SQL		operators	231
Blocks	203	UTU1 - Extraction of 'P' lines with	
TRDQ - Introduction.	203	'UNS'	231
TRDQ - User Input	204	UTU1 - Input / Processing / Results	231
TRDQ - Description of Steps	205	UTU1 - Description of Steps	231
TRDQ - Execution JCL	207	UTU1 - Execution Script	231
		UTU2 - Update of 'P' lines with 'UNS'	232
Chapter 9. Components	211	UTU2 - Input / Processing / Results	233
Server Environment Components	211	UTU2 - Description of Steps	233
Introduction.	211	UTU2 - Execution Script	233
On-Line Documentation.	211	UTM1 UTM2 - Conversion of 'old' Meta	
Generation Skeletons.	211	Entities	234
Administration Database	212	UTM1 - Description of Steps	237
Administration Database Files	212	UTM1 - Execution Script	237
Administration Database Backup.	212	UTM2 - Input / Processing / Results	238
Development Database	212	UTM2 - Description of Steps	238
Development Database Files	212	UTM2 - Execution Script	239

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

Bibliography

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

Chapter 2. Prerequisites

Prerequisites for the Server Environment

Hardware and Software

- Architecture: a Windows 2003 server or a Windows 2008 server.
- Processor: workstation with Windows Server 2003 32 bits or Windows Server 2008 32 bits.
- 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 onwards. You can download it from the following URL:
www.microsoft.com/downloads
 - For the MICRO FOCUS version:
MICRO FOCUS Application Server 3.1, 4.0 or higher
 - For the Acucorp version:
ACUCOBOL-GT 7.0

CAUTION:

Besides installing the COBOL runtime, you must update the system environment variables (PATH, COBPATH...).

For ACUCOBOL-GT, the PATH system variable must be completed with the path of the runtime (usually ..\AcuGT\bin). The machine must then be rebooted.

- Installation medium : 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	54
Total for installation test user files	35

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

Prerequisites for the Client Environment

Hardware

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

- Processor: Intel Pentium III 450 Mhz minimum or compatible processor.
- Monitor: graphic monitor (800x600) VGA or higher resolution (XGA or SVGA).
- CD-ROM drive.
- Card: adapted to the site network.
- Memory (RAM): 512 Mb (1 Gb advised).

Disk Space

Required disk space:

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

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

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

Software

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

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

Communications

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

Chapter 3. Installation of the 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 disk drive 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 the database data is stored. 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 conversion file, the file which contains the error or information messages output by the procedures,
- 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 load-modules for the interface between VA Pacbase and DSMS 2.5 are provided in standard; for a DSMS use, the DEXP procedure must be adapted to use these new load-modules instead of PDS600 and PDS610 which have no longer been compatible since VA Pacbase 3.0.

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 startup 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\[Database_name]\TMP\[user_code]\[proc_name]-[number]`
They are removed at the end of the procedure execution. If you do not want these files to be deleted, you must previously set the "DelDo" variable to "NO". This setting can be done in two ways:
 - either in the startup script with the `WshEnv(DelDo)="NO"` command. In this case, the variable is valid for the procedure only.
 - either in the Init.vbs file with `DelDo = "NO"`. In that case, the variable is valid for ALL the procedures.

In both cases, you will have to delete the temporary files manually.

- 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 automatically. These files (16 in total, with a name which begins with 'T') 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) and the supervisor (PRBVP.vbs):

- via a command line,
- by double-clicking on them,
- via the 'Start' menu. Access the shortcut: [[dbname] Database Utilities] in the [VisualAge Pacbase Server] group of [dbname] 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: [dbname] [script_name] ; [dbname] is located under directory \DATA.

To be executed, each procedure (whose name is represented by xxxx) requires the definition of various elements:

1. either in the launch script: This launch script, BVPxxxx.wsf, is in Windows Scripting.
 - a. User code,
 - b. Database external name (directory under \DATA),
 - c. 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).
 - d. 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.
 - e. 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.

- f. The debug mode can be activated for the procedure by setting 2 variables: DelDo used to keep the temporary file and BVPTrace to activate the Trace mode. These variables can be set independently:

```
WshEnv("DelDo")="NO"
```

```
WshEnv("BVPTrace")="YES"
```

In this case, the variables are set for the procedure only.

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: [dbase] and [user]

- b. Assignment of 2 parameters with 2 possible values:

DelDo = 'YES'/'NO': the temporary files are deleted (YES) or not (NO).

Trace = 'YES'/'NO':

The BVPTrace parameter has a default value set to 'NO' in the init.vbs file. In this case, only the COBOL 'displays' are written in the DISPxxxx.log file.

If the BVPTrace parameter is set to 'YES', the VA Pac context (list of the directories in use) is written in the xxxxMSG.log file.

With Acucobol, the trace is activated and the execution errors are written in the acutrace.txt file.

With Microfocus, the execution errors are written in the xxxxMSG.log file ; some more details may be written in the DISPxxxx.log file, especially input-output errors on files.

Moreover, with a Microfocus runtime greater than version 4.0, you can activate the Consolidated Trace Facility (CTF). This trace is activated for all the programs indicated in the mftrace.cfg configuration file, located in the \$INSTDIR\Server\Config\Ctf directory.

You must modify this configuration file to indicate, among others, the program code (BVPxxxx) that you want to trace in the mftrace.application parameter. The result of the trace, bvprtrace.txt, is automatically computed when the procedure is run and is located in the directory which contains the execution report.

So you are advised against modifying the mftrace.emitter.textfile parameters.

The information related to the parameters of this configuration file can be found at:

<http://supportline.microfocus.com/Documentation/books/sx40sp2/sgpubb.htm>

(Click on 'Configuration File Entries').

- 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 start-up script, it is searched for in the input file.

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

Notes:

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.

Now the PC file (Database backup) is defined by an environment variable: WshEnv("BVP_SaveName").

Via this variable, the parameter for the definition of assignments (PAC7PC, PAC7PD, PAC7PY) is created. The resulting 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:** 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 set to blanks) and the 'ADMIN' user code (with the password 'ADMIN'). These user codes will be used to activate the keys and to launch the test scripts.
- **REST:** restoration of the test Administration database
- **VINS:** installation of the Administration Model

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 be declared in the Administration Database.

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

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

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

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

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 chapter 4 'Installation/Re-installation of Client Components', sub-chapter 'Fundamentals of VA Pac Client-Server Communication').

- 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: declaration of the Development Database in the Administration Database,
- REST: restoration of the Test Database,
- VINS: installation of the Development Database Model.

Access key entering

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

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

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

Backup of the Administration Database

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

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 'Delete [database_name] Database' shortcut located under the [VisualAge Pacbase 3.5 Server] programs group in the Start menu enables you to delete a Windows database on a Windows platform.

This procedure executes the following operations:

- deletion of the Windows service for a given Database.
- deletion of configuration information (CONFIG directory).
- deletion of information specific to the Database in the Windows registry database.
- 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, online programs and their generation characteristics.
- In case of system operation problem, this list should be printed in order to communicate to VisualAge Pacbase Support all the installation references.

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

INSL - Execution Script

```
| -----  
| VISUALAGE PACBASE  
| -----  
| - LIST OF INSTALLED PROGRAMS AND FILES -  
| -----  
|  
| -----  
|  
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 = " BVPSEVER.EXE BVPDIAL.EXE BVPWSH.OCX "
ListExclude = " BVPLOCK.DLL BVPPAUSE.DLL " & ListExclude
ListExclude = " BVPSPLIT.EXE BVPTRANS.EXE " & ListExclude
ListExclude = " CODEPAGECONV.DLL VAPUTIL.DLL " & ListExclude
ListExclude = " BVP.ICO BVPSEVER.ICO " & ListExclude

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

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

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

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

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

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

```

```

Return = WshShell.Run( sCmd, 10 , TRUE)
End If

' End of job :
'-----
If ( Return <> 0 ) Then
    Display = "Abnormal end of job " & _
    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)

```

Connection

Starting and Stopping a Server

The 'Start [dbase_name] Database Service' shortcut located under the [VisualAge Pacbase 3.5 Server] Programs group of the Start menu enables you to launch the listener on the [dbase_name] database.

The 'Stop [dbase_name] Database Service' shortcut located under the [VisualAge Pacbase 3.5 Server] Programs group of the Start menu enables you to stop the listener on the [dbase_name] database.

The operation parameters of the listener are defined in the 'Server.wsf' procedure.

When abnormal executions occur in a listener, messages can be displayed in the list of events of the 'Event Viewer' administration tool, under 'Log/Application'.

These 'ERROR'-category events can give a first indication. They generally correspond to problems in the execution environment.

However, in case of abnormal processing, the product support may ask you to activate the 'trace mode' to detect the source of the error.

- TRACE MODE

Different trace levels can be implemented:

- Level 1

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

- Level 2

Detailed trace of the listener processing,

- Level 4

Trace of the messages between the listener and the client workstation.

In the listener startup procedure, 'server.wsf', the purpose of the SRV_TRACE environment variable is to activate the trace mode. To use another trace level, you must set the SRV_TRACE variable and re-start the listener.

EXAMPLE:

SRV_TRACE=1 for a trace level 1

SRV_TRACE=3 for a trace level 1 and 2

SRV_TRACE=5 for a trace level 1 and 4

There are two types of trace result files:

- srv[process_number].txt

to trace the listener (BvpServer.exe).

- dial[process_number].txt

to trace each listener connection (BvpDial.exe).

These files are located in the SRV_DIR directory, set by default in server.wsf file to:

.../data/[database_name]/tmp

The SRV_TRACE_DEL environment variable is set to keep all or a part of the traces created by the listener execution in "dialnnn.txt" :

SRV_TRACE_DEL : "ON" (default value)

to keep traces produced by a processing error only.

SRV_TRACE_DEL : "OFF"

to keep all the traces.

Database under DSMS control

If the [database name] Database is under DSMS control, the DSMS files must then be declared when the listener of the [database name] Database is started up.

To do so, you must validate the lines provided for this purpose in the "Server.wsf" procedure and enter there the name of the DSMS database installed on your workstation.

Connecting to a 3270 Emulator

You can access an online server in a 'dumb terminal' mode via a 3270 emulator.

To access a Database, in a 3270 mode via an online server, you must configure the emulator by indicating:

- the IP address of the machine where the online server is installed,
- the online server port number, chosen at installation time when the Database is created.

The code page of the emulator must be valorized according to the Database language code:

- code page 1147 for a French Database,
- code page 1146 for an English Database.

These code pages are automatically set, in the "Server.wsf" procedure, when the online server is started up.

Monitor for the Management of Listeners

A tool, located in the SYS/PGM directory, enables you to manage the listeners.

This utility allows to display the status of the clients connected to a server identified by the socket number in use.

You launch this tool by executing:

```
TP.exe <SocketN°_Server>
```

When executed, this command displays, among others:

- the PID,
- the IP address,
- the program being executed,
- the connected user,
- the connection time.

Chapter 4. Installation/Reinstallation of Client Components

Things to Know Before Installing

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

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

Root Directory

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

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

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

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

NOTE: Do not use blank characters in directory names.

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

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

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

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

Installation Startup

Insert the installation CD-Rom.

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

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

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

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

Choose the Client component you want to install.

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

Fundamentals of VA Pac Client-Server Communication

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

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

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

AVAILABLE COMMUNICATION TYPES

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

MIDDLEWARE

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

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

However you can also choose a communication via a gateway.

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

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

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

COMMUNICATION FILES

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

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

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

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

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

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

THE VAPLOCAT.INI FILE

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

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

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

A location :

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

THE BASES.INI FILE

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

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

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

Administrator & Developer workbench

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

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

- Administrator workbench
- Developer workbench

Developer workbench includes the following modules, each running independently:

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

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

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

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

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

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

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

To do this, a window enables you to enter:

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

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

2. The Database logical code.

Maximum length: 4 characters.

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

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

3. The location name

Maximum length: 20 characters.

Default: Location-1

More than one Database may be associated with one location.

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

You select the mode via two check boxes.

VisualAge Pacbase signon:

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

Middleware signon:

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

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

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

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

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

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

START-UP FILES

The start-up files are :

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

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

START MENU / PROGRAMS CHOICE

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

Administrator-Developer workbench

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

Open Jade and Tidy / Publishing facility

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

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

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

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

eBusiness Tools

The eBusiness Tools are:

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

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

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

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

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

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

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

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

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

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

START-UP FILES

The start-up files are :

- For the Proxy Generator:

vapGen.exe

- For the Location Editor:

vapLocationEditor.exe

- For the Services Test Facility:

vapServicesTestFacility.exe

- For VisualAge Pacbase Web Connection:

PacWebgen.exe

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

ebusinesstools\bin

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

START MENU / PROGRAMS CHOICE

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

eBusinessTools

Location Editor

Proxy Generator

Services test Facility

Dialog Web Revamping Generator

VisualAge Pacbase WorkStation

WARNING: The WorkStation runs on a 16 bits environment; check that the WorkStation is installed in an environment compatible with such applications.

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

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

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

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

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

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

In the next panel, select the elements to install:

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

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

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

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

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

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

To do this, a panel enables you to enter:

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

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

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

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

2. The Database logical code.

Maximum length: 4 characters.

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

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

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

3. The location name

Maximum length: 20 characters.

Default: Location-1

More than one Database may be associated with one location.

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

You select the mode via two check boxes:

'Pacbase is secured':

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

'Middleware is secured':

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

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

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

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

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

START-UP FILE

The start-up file is :

pexec.exe

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

START MENU / PROGRAMS CHOICE

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

WorkStation

- WorkStation 3.5
- WorkStation 3.5 News
- <methodology> News

INSTALLATION PARAMETERS FILE

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

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

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

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

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

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

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

DESCRIPTION OF THE PARAMETERS FILE

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

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

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

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

The Pacbase.dat file should not be destroyed.

The possible values for the Methodology parameter are:

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

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

WARNING ON VAPLOCAT.INI FILE

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

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

FILES REQUIRED FOR THE USE OF THE GENERATION-PRINT MONITOR

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

This file is saved under the following path:

SPAC\NNNL\GPMON.PRM

If you do not find this file in the directories tree, you must create it in the directories indicated just before. You just have to name it GPMON and to assign it a .PRM extension.

Then, open the file under Notepad, for example, and enter an 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 the labels of standard files produced by a customized generation-print procedure, you need to create a GPMON.LBL file in which you enter the specific labels of these files.

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

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

Web Application Models (WAM)

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

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

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

Middleware

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

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

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

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

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

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

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

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

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

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

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

LIST OF PARAMETERS

- VAP SOCKET
 - IP address: IP address and port used by the VA Pac server.
Do not enter left-aligned '0', source of error with some configurations.
- MVS CICS SOCKET
 - IP address: IP address and port used by the VA Pac server.
Do not enter left aligned '0', source of error with some configurations.
 - Transaction code: Code of the CICS transaction of the VA Pac Communication Monitor.
 - Code Page: Value identifying the coding of characters used by the VA Pac server.
1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS CICS CPI-C
 - Destination-id entry: BVPSCPI (default value).
If you modify this value, it must be the same as the value entered in the Symbolic destination name, a parameter included in the configuration of this communication protocol.
 - Code page: Value identifying the coding of characters used by the VA Pac server.
1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MQ-CICS Bridge and MQ-IMS Bridge
 - LocCcsid: Character-set code of the machine in local (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.
Maximum: 9 characters.
 - Queue Manager name: Name of the local Queue Manager of the client part.
If there is an intermediate MQSeries Server on NT, this parameter must identify the intermediate NT Queue Manager and not the Queue Manager.
Maximum: 4 characters.
 - Request Queue name:
Maximum: 48 characters.
 - Reply Queue name:
Maximum: 48 characters.
 - Transaction code:

Maximum of 4 characters for CICS.

Maximum of 8 characters for IMS.

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

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

Editing Communication Parameters

The bases.ini File

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

By default this file's access path is:

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

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

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

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

NOTE: With the VA Pac WorkStation, you may use a customized parameters file. To do so, enter -- after the VA Pac Database name -- this file's

name framed by the '<' and '>' signs. Complete details on these parameters are given above, at the end of the 'VisualAge Pacbase WorkStation' subchapter.

DESCRIPTION OF A SECTION'S CONTENTS

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

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

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

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

The following parameters vary according to the chosen option.

- connectionCleaningInterval = period of time (in millisecond) between two startups of connections cleaning procedures.
Parameter needed for the VisualAge Pacbase Workstation only.
Default value for Windows NT/2000/XP = 60000.
For Windows 95/98, this parameter must be set to -1, and cannot be modified, to avoid problems of multi-threading, not supported by these platforms.

PARAMETERS FOR DIRECT ADAPTER (LOCAL MIDDLEWARE)

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

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

- location = location name for the Database

Maximum length: 20 characters.

Default: Location-1

More than one Database can point to the same location.

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

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

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

- traceLevel = trace level of the middleware execution. Its possible values are:

- 0 : no trace
- 1 : trace with errors (default)
- 2 : standard trace, not detailed
- 3 : trace for information
- 4 and + : trace for debugging

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

Default: ..\middleware\CharConv.txt

PARAMETERS FOR GATEWAY ADAPTER (REMOTE MIDDLEWARE)

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

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

Default: 127.0.0.1 for a local host

- port = value of the IP port where the gateway receives the client requests.

Default: 5647

- location = location name for the Database

Maximum length: 20 characters.

Default : Location-1

More than one Database can point to the same location.

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

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

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

The vaplocat.ini File

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

By default, the path to this file is:

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

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

The maximum length of this name is 20 characters.

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

- VAP SOCKET

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

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

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

```

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

```

- IMS Connect

```

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

```

- MQ-IMS Bridge

```

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

```

DETAILS ON THE PARAMETERS

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

- COMM_TYPE:

This parameter identifies the communication protocol in use.

The possible values are:

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

- TCPMVS: VA Pac Server under MVS/CICS with the use of a TCP/IP listener.
 - CPIC: VA Pac Server under MVS/CICS, with the use of the CPI-C protocol.
 - MQCICS: VA Pac Server under MVS/CICS, with the use of the MQ-CICS-BRIDGE protocol.
 - MQMCICS: Same as MQCICS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.
 - TCPIMS: VA Pac Server under MVS/IMS, with the use of the IMS Connect protocol.
 - MQIMS: VA Pac Server under MVS/IMS, with the use of the MQ-IMS-BRIDGE protocol.
 - MQMIMS: Same as MQIMS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.
- IXO_ADDRESS: IP address and port used by the VA Pac Server.
Do not enter left-aligned '0', source of error with some configurations.
The port number must correspond to the one indicated at the host machine configuration for OS390 CICS and IMS/VS.
The same for Windows or UNIX, but for these platforms, a number specifically identifies a VA Pac Database. For more details, refer to the 'Installation of Server Environment' chapter, 'Repository Installation' subchapter, 'Development Databases' section, in the respective Installation Guides.
 - IXO_DATASTORE: Name of link to IMS defined in IMS Connect (IMS DataStore ID).
 - IXO_DYNAMICREPLYQUEUE
MQ-CICS Bridge and MQ-IMS Bridge.
When this parameter is set, the reply Queue is dynamically created by MQSeries to pass the response messages (consult the MQSeries documentation to use this type of Queue). This parameter must contain the dynamic name of the Queue (Dynamic Queue name) and the IXO_REPLYQUEUE parameter must contain the 'Queue Model' name (48 characters maximum).
 - IXO_HEADERVERSION: Version of the MQCIH structure (for MQ-CICS Bridge) or of the MQIIH structure (for MQ-IMS Bridge).
1 (default) or 2, depending on the Bridge interface in use (1 character maximum).
 - IXO_LOCALCCSID: code of the character set of the machine locally
MQ-CICS Bridge and MQ-IMS Bridge.

This code (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.

(9 characters maximum).

- IXO_QUEUEMANAGER: Local Queue Manager name of the client part. MQ-CICS Bridge and MQ-IMS Bridge.

(4 characters maximum).

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

- IXO_RACFGROUP: Name of RACF group for IMS Connect.

- IXO_REPLYQUEUE: Name of reply messages Queue.

MQ-CICS Bridge and MQ-IMS Bridge.

(48 characters maximum).

- IXO_REQUESTEXPIRY: Expiration delay of the request message, in seconds.

MQ-CICS Bridge and MQ-IMS Bridge.

(9 characters maximum).

- IXO_REQUESTQUEUE: Name of request messages Queue.

MQ-CICS Bridge and MQ-IMS Bridge.

(48 characters maximum).

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

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

- IXO_TRANSID: Transaction code.

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

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

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

Component Updating, Modification, or Removing

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

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

This option is available only with:

- Administrator & Developer workbench:

To add one of both workbenches as the other is already installed,

- VisualAge Pacbase WorkStation:

To add a Methodology.

- 'Remove' a component which is already installed.

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

Chapter 5. Tests

List of Utilities

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

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

Installation Tests

The VA Pac Installation tests include the following operations:

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

Generation-Print, Online and Batch Update Tests

These tests consist of the following steps:

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

Administration Database Procedures Tests

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

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

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

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

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

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

Development Database Procedures Tests

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

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

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

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

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 proposed if you have an older version already installed.

The operational startup scripts, in the \DATA\[Database_name]\SCRIPT directories, are not impacted. To get the last version of the scripts, upon the 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:

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

Winnt\Profiles\AllUsers\Start Menu\Programs\VisualAge
Pacbase 3.5 Server\[database_name]

- Delete the key from the HK_Local_Machine register:

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 BVPDELDDB 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 complemented by the execution of utilities procedures.

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

There are two types of procedures:

Procedures to be executed on the 2.5 Database.

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

These procedures are :

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

Procedures to be executed on the new Database.

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

These procedures are :

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

Notes

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

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

Example :

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

```
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.

Sequence of operations

It consists of seven steps:

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

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

2) Installation of the Administration Database

To install the Administration Database, execute the installation process.

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

You must execute the following procedures:

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

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

3) Retrieval of a Development Database.

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

You must execute the following procedures:

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

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

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

Steps 4, 5, 6 and 7 are optional.

4) Retrieval of generation-print commands.

It consists in executing the following procedures:

- PG20: if retrieval of 2.0 PG file,

- PG25: if retrieval of 2.5 PG file.

5) Retrieval of PJ journal

It involves executing the following procedure:

- PJ25: retrieval of PJ file data.

6) Retrieval of PEI files.

It involves executing the following procedure:

- PP25: retrieval of PP file,

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

It involves executing the following procedure:

- UV25: retrieval of UV file data.

Retrieval of User Parameters (PE25)

PE25 - Introduction

Principle

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

Execution conditions

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

Printed output

This procedure prints a report which indicates the errors encountered.

Result

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

Notes

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

The retrieval reinitializes the license assignments.

PE25 - Input / Processing / Results

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

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

The 'A' line has the following structure:

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

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

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

The 'B' line has the following structure:

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

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

The 'C' line has the following structure:

Position	Length	Value	Meaning
2	1	'C'	Line code

Position	Length	Value	Meaning
3	4	bbbb	Database code

PE25 - Description of Steps

Processing of user parameters (PE): PTU920

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

Transaction formatting: PAF900

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

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

Administration Database update: PACA15

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

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

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

Return codes:

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

PE25 - Execution Script

```

'-----
'          VISUALAGE PACBASE
'-----
'          RETRIEVAL OF PE FILE
'-----
'
<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

' No Rollback
WshEnv("BVPRB") = "N"

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("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 = 2 Then
Call Msg_Log (Array("1061"))
End If
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>

```

Retrieval of the Development Database (PC25)

PC25 - Introduction

Principle

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

Execution conditions

None.

Printed output

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

Result

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

PC25 - Notes on Data Retrieval

Splitting up of the comment description (-G)

The comment description is split up into several descriptions.

- Comments

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

Caution

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

- Generation lines

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

- Generation parameters

They include the O line type (-GO).

- Error messages management

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

- Call of entities via Relations

They include the R line type (-CR).

- Specificity of the Input Aid entity

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

WARNING:

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

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

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

Data structures table type

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

Transformation of U type manuals

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

Extension data: User entities, Input Aids, Report Layouts

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

PC25 - Input / Processing / Results

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

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

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

This line must be structured in this way:

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

PC25 - Description of Steps

General processes: PTU911

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

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

Manuals and volumes processing: PTU909

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

Comments processing: PTU92A

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

Sort of Input aids : PTU92B

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

Sort of Input aids : PTU92C

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

Report layout processing: PTU919

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

Meta entities processing: PTU912

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

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

User entities processing: PTU913

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

Sort of long data: PTU91A

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

Files merging: PTU914

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

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

Code	Physical name	Type	Label
PAC7PO	Tmp dir. : WPO	Input	Comments (no call of Input Aids)
PAC7PD	Tmp dir. : 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

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          RETRIEVAL OF PC FILE      |
|-----|
|
<job id=PC25>

<script language="VBScript">
Dim MyProc
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 & "\WXZ.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 & "\WXZ.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

Principle

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

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

Execution conditions

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

Printed output

This procedure outputs a report which contains the errors encountered.

Note

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

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

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

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

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

PG20 - Input / Processing / Results

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

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

The line structure is as follows:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
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. Database - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGY or WGZ	Input	Update transactions

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

Administration Database update: PACA15

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

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

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

Return codes:

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

Transactions formatting: PAF900

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

Development Database update: PACA15

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

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

Return codes:

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

PG20 - Execution Script

```

|-----|
| VISUALAGE PACBASE |
|-----|
| RETRIEVAL OF PG FILE SINCE 2.0 |
|-----|

```



```

'
<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

' No Rollback
WshEnv("BVPRB") = "N"

Call Msg_Log (Array("1022" , "PTU908"))
'-----
Call BvpEnv("PTU908","PAC7IN",Rep_SAVE & "\PG20")
Call BvpEnv("PTU908","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("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 = 2 Then
Call Msg_Log (Array("1061"))
End If
If Return = 4 Then
Call Msg_Log (Array("1060"))
End If
If Return > 4 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("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 = 2 Then
Call Msg_Log (Array("1061"))
End If

```

```

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>

```

Generation-Print Commands Retrieval (PG25)

PG25 - Introduction

Principle

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

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

Execution conditions

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

Printed output

This procedure prints a report on the errors encountered.

Note

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

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

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

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

PG25 - Input / Processing / Results

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

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

The line structure is as follows:

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

PG25 - Description of Steps

Generation-print commands processing: PTU921

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

Transaction formatting: PAF900

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

Administration Database update: PACA15

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

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

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

Return codes:

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

Transactions formatting: PAF900

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

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

Development Database update: PACA15

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

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

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

Return codes:

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

PG25 - Execution Script

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          RETRIEVAL OF PG FILE SINCE 2.5          |
|-----|
|
<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

```

```

' No Rollback
WshEnv("BVPRB") = "N"

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")

If Return <> 0 then
Call Msg_Log (Array("1025", "PAF900/ADMinistration", Return))
End If

If Return = 0 then
Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
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("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 = 2 Then
Call Msg_Log (Array("1061"))
End If
If Return = 4 Then
Call Msg_Log (Array("1060"))
End If
If Return > 4 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 ("BVPAPAF900")
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("PAC7RB") = Rep_TMP & "\WRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\WRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
If Return = 2 Then
Call Msg_Log (Array("1061"))
End If
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>

```

Retrieval of PJ Transactions (PJ25)

PJ25 - Introduction

Principle

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

Execution conditions

None.

Printed output

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

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

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

Result

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

Comments

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

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

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

PJ25 - Description of Steps

Processing of PJ transactions sequential file: PTU918

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

Code	Physical name	Type	Label
PAC7ET	User dir.: PJ25ET918	Report	Report

PJ25 - Execution Script

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          RETRIEVAL OF PJ FILE      |
|-----|
|
|<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>

```

Retrieval of PP file (PP25)

PP25 - Introduction

Principle

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

Execution conditions

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

Printed output

This procedure outputs a report which indicates the error encountered.

Result

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

Note

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

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

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

PP25 - Input / Processing / Results

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

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

The structure of the line is presented as follows:

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

PP25 - Description of Steps

Management of production environment: PTU923

Code	Physical name	Type	Label
PAC7PP	Save dir. : OLDPP	Input	Back up of production environment (old release)
PAC7AE	System - Skel. dir. : AE	Input	Error message
PAC7MB	User input	Input	User input
PAC7AR	Base dir. : AR	Input	Development Database data
PAC7AN	Base dir. : AN	Input	Development Database index
PAC7AY	Base dir. : AY	Input	Development Database extension data
PAC7GY	Tmp dir. : WGY	Output	Records of production environments (length=310)
PAC7ET	User dir. : PP25ET923	Report	Report in case of error

Transactions formatting: PAF900

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

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

Development Database update: PACA15

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

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

Return codes:

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

PP25 - Execution Script

```
'-----  
'          VISUALAGE PACBASE  
'-----  
'          RETRIEVAL OF PP FILE  
'-----  
'  
<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 & "\OLDPPP")  
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 ("BVP900")
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("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 = 2 Then
Call Msg_Log (Array("1061"))
End If
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>

```

Retrieval of passwords (UTMP)

UTMP - Introduction

Principle

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

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

Execution conditions

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

Printed report

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

Result

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

UTMP - Description of Steps

2.N passwords retrieval: PTURMP

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

UTMP - Execution Script

```
|-----|
| VISUALAGE PACBASE |
|-----|
|          PASSWORD RETRIEVAL          |
|-----|
|
|<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. Database - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users

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

Update of the Administration Database: PACA15

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

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

Return codes:

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

UV25 - Execution Script

```

|-----|
|      VISUALAGE PACBASE      |
|-----|
|           RETRIEVAL OF UV FILE           |
|-----|
|
<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

Principle

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

Limits

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

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

MB25 - Description of Steps

MB file retrieval: PTU926

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

Code	Physical name	Type	Label
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7MB	User dir.: OLDMB	Input	2.5 MB transactions
PAC7MV	User dir.: MB25MV	Output	Retrieval transactions for UPDT
PAC7ME	User dir.: MB25ME	Output	Revoked transactions
PAC7EF	User dir.: MB25EF926	Report	Retrieval reports
PAC7DD	User dir.: MB25DD926	Report	Batch procedure authorization option

MB25 - Execution Script

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          RETRIEVAL OF MB FILE
|
| -----
|
<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

Principle

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

Limits

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

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

GY25 - Description of Steps

GY file retrieval: PTU927

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

GY25 - Execution Script

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          RETRIEVAL OF GY FILE
|
| -----
|
<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

Principle

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

Limits

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

MB30 - Description of Steps

MB file retrieval: PTU928

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

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          RETRIEVAL OF MB FILE
|
| -----
|
<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

Principle

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

Limits

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

GY30 - Description of Steps

GY file retrieval: PTU929

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

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          RETRIEVAL OF GY FILE
|
| -----
|
<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("PAC7AN") = Rep_BASE & "\AN"
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
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
GPMC	Generation of MOVE CORRESPONDING
GPRC	Generation (COBOL API)
REOR	Re-organization of Administration Database
REST	Restoration of Administration Database

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

Procedure	Comments
UPGP	PAF update of Administration Database

List of procedures suppressed since 2.5 release

Procedure	Program	Comments
CPSN	PTU850 PTU855	Integrated into PACX procedure
EMSN	PTU810	
MESN	PTU815	
SASN	PTU130 PTU140	Integrated into PACS procedure
MLIB	PTU100 PTU120	Integrated into PACS procedure
SAVE	PTU500	Integrated into PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACT11	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PREI986	
RVKE	PREI40 PREI50	
STOP	PTUR00	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	
VPU2	PVA320	

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

Retrieval of VisualAge Pacbase 3.0

Operations to be Performed

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

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

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

Chapter 8. RPP Utilities

MIAM - Migration Help Function

MIAM - Introduction

Principle

This procedure provides the following functions:

- Search for special characters: loading of the special characters file (CS file).
- Search for entities whose code corresponds to a string not allowed (not compatible with Windows for example): loading of the special characters file (CS file).
- Search for Segments renamed as 00 in the -CD when the Segment 00 exists in the network.
- Search for Reports with duplicate labels.
- Search for gaps in the sequence of label numbers of Reports.
- Search for Structures called in Categories of a Report and not defined in the description of the Report.
- Search for Report structure lines calling several Data Elements with the same position.
- Search for labels called in Report categories and not defined in the description of the Report.
- Search in the -GE: references to Segments or Data Elements that do not exist in the -CE or -CS of Screens, or reference to a Text that does not exist.
- Search for Standard or Client Screens, other than Help screens, that have a generation variant different from N and that only have a definition or -CE lines.
- Search in the Screen -CS: multiple uses of the same Segment in display or reception in the same category.
- Search for Macro working lines that declare a table whose name contains the \$ character.
- Search for working lines not to retrieve (with a type M, C, L or A).
- Search for obsolete generation variants.
- Search for Screens and Programs (not typed as Macros or not used as Macros) generated with the same external name.
- Search for duplicate map names for Standard Screens or Clients that have a generation variant X0, X1, X2, X3, 00, 01, 02, 03, 4F, 80, 10, or 12.
- Search for the uses of Model entities in standard entities.

- Search for EL-type function lines with a level not compatible with a CO-
DU- DW-type function that precedes them.
- Search for orphan conditions: condition lines with a type AN OR DU DW
DC DI DV coming from a Macro and preceded by an IF-type condition line
that does not come from a Macro.
- Search for duplicate P-lines with an N type.
- Search for P-lines of Macro with a function level lower than 99 associated
with an operator code different from 'N', '*', 'SUP' and '\$n'.
- Search for P-lines of Macros, assigned or not, that contain a non-numeric
line number.
- Search for P entities to be transformed into Macros: loading of PM file.
- Search for Programs not yet taken into account (Type of COBOL C, or Type
and structure D, F, S).
- Search for Programs coming from PacReverse.
- Search in standard, client or server Screens, for P lines that contain 99BL,
99IT, 99DV, 99DI, 99DC preceded in the same function/subfunction by a
comment line with an OR- or AN-type.
- Search for entities not to retrieve: loading of the NR file used by the MIBR
and MIBJ procedures. This applies to:
 - Data Elements with a usage not recognized in RPP (different from 0 1 2
3 5 6 7 8 9 C D F G H I J N O P Q R T U W X Y Z); Data Elements with
a usage 4 and a Large Object type (L-type) are exceptions and are
recognized,
 - Database Blocks with a type AR, D1, D3, QA, QD, QG, QI, QO, QV, SE,
TI, TR
 - Reports for Layouts.
 - Bulk-type Meta Entities and the associated User Entities
 - Pacdesign Meta Entities and the associated User Entities as well as the
Data Elements that define the 7M 7N 5Q Meta Entities if they have not
been retrieved with the MT parameter in the BVPAMIAM file.
 - F-type Dialogs or Screens (Folder Server) or FV (Folder View)
 - Dialogs or Screens that contain PAF commands (EXP operator) if the PF
parameter with a value NO was indicated in the BVPAMIAM parameter
file.
 - Programs with an obsolete type and structure (value C).

For each case, a record is written in the RQ file of errors.

Description of CS file of special characters or strings that are detected:

Pos.	Len.	Value	Meaning
1	2		Type of character
		WN	Character not compatible with WINDOWS
		CS	Special character
3	6		Detected character or string
9	6		Substitute character or string: this field must be filled in before the execution of the MIBA or MIBR procedures
15	66		FILLER

Description of NR file of entities not to retrieve:

Pos.	Len.	Meaning
1	3	Library code
4	3	PAF type
7	30	Entity code
37	2	Block type
39	1	Data Element usage
40	6	Meta Entity
46	35	FILLER

Description of NS file of Screens with only a definition or -CE lines

Pos.	Len.	Meaning
1	3	Library code
4	3	PAF type
7	30	Entity code

Description of PM file of non typed Macros

Pos.	Len.	Meaning
1	6	Entity code
7	44	FILLER

Description of RQ file of errors:

Pos.	Len.	Value	Meaning
1	3		Library
4	3		PAF type
7	6		Entity code
13	24		FILLER
37	3		Error type
		BL	Block not to retrieve
		CS	Special character
		DI1	Dialog with an F type (Folder) or an FV type (Folder View)
		EC2	Screen with an F type (Folder) or an FV type (Folder View)
		EC3	Screen whose -CS contains multiple uses of the same Segment in display or reception in the same category
		EC4	Standard Screen or Client with a generation variant different from N, that contains only a definition or -CE lines
		EC5	P-line with 99BL, 99IT, 99DV, 99DI, 99DC, preceded by a comment line with an OR- or AN-type
		ET1	Report with duplicate label
		ET3	Report for PDM layout
		ET4	Non existing label called in a category of the Report
		ET5	Non existing structure called in a category of the Report
		ET6	Structure that calls several Data Elements with the same position
		ET7	Missing label in the labels sequence of the Report
		EU1	Pacdesign User Entity not retrieved
		EU2	Bulk-type User Entity not retrieved
		GE1	Non-existing reference (-GE)
		MCN	Macro with no type (indicating PARAM or NO PARAM)
		MP4	EL-type function whose level is not compatible with a preceding CO- DU- or DW-type function
		MP5	Non-numeric line number
		MT1	Pacdesign Meta Entity not retrieved
		MT2	Bulk-type Meta Entity not retrieved

Pos.	Len.	Value	Meaning
		MT3	Data Element of Meta Entity not retrieved
		NEX	Duplicate name of generated entity
		MEX	Duplicate name of generated MAP
		NV1	Function level not compatible with the operator
		OR1	Orphan conditions
		PA1	Presence of PAF operator
		RU1	Use of Model entity in standard entity
		PG1	Duplicate P-line with N type
		PG2	Program with an obsolete type and structure
		PG3	Program with a variant or type and structure not yet taken into account
		PG4	Program coming from PacReverse not taken into account
		SG1	Segment renamed in -CD
		UR	Data Element usage not to retrieve
		VAR	Obsolete generation variant
		WK	Working line with a type M C L or A
		WK1	Macro working line that declares a table whose name contains a \$ character
		WN	Characters not compatible with WINDOWS
40	40		Error label
80	2		Section
82	2		Paragraph
84	3		Line number
87	2		File code
89	4		Segment code
93	2		Label number
95	6		Volume code
101	6		Data Element code
107	3		PAF type
110	6		Entity code
116	2		User Entity call type
118	3		Screen type (1)
121	30		FILLER

(1) The Screen type is only specified for errors detected on the definition of the Screen, with the value SRC for a standard Screen, CLI for a Client Screen, SRV for Server Screen and, ECR for the other Screens.

Consequences and actions to be taken for the different error types.

Code	Meaning	Consequence	Possible action	Consequence if no action
CS	Special character	Population of the CS file	(1)	TA2 import rejected. Generation error if the entity is used.
WN	Character not allowed (ex: not compatible with Windows)	Population of the CS file	(1)	TA2 import rejected. Generation error if the entity is used.
MCN	Non typed Macro	Population of the PM file	(2)	
VAR	Obsolete generation variant	Error in the RQ file	Populate the NR file	Generation rejected
EC3	Multiple use of the same Segment in display or reception in the same category of the -CS of a Screen	Error in the RQ file	Correct in VA Pacbase	Generation error
EC5	P line with 99BL, 99IT, 99DV, 99DI, 99DC preceded by a comment line with an OR- or AN-type	Error in the RQ file	Correct in VA Pacbase	Dot not generated before the line
ET1	Report with duplicate label	Error in the RQ file	Correct in VA Pacbase	Warning in RPP
ET4	Non existing label called in a category of the Report	Error in the RQ file	Correct in VA Pacbase: remove the call to the label	Generation difference in the migration help
ET5	Non existing structure called in a category of the Report	Error in the RQ file	Correct in VA Pacbase: remove the call to the structure	Generation difference in the migration help

Code	Meaning	Consequence	Possible action	Consequence if no action
ET6	Structure that calls several Data Elements with the same position	Error in the RQ file	Correct in VA Pacbase: correct the structure	Generation difference in the migration help
ET7	Missing label in the labels sequence of the Report	Error in the RQ file	Correct in VA Pacbase: correct	Generation difference in the migration help
GE1	Non-existing reference (-GE)	Error in the RQ file	Correct in VA Pacbase: modify the -CE or -CS or the call to text reference	Error reported in the TA2 import (change into comments)
MP4	EL-type function whose level is not whose level is not compatible with a preceding CO-DU- or DW-type function	Error in the RQ the RQ file	Correct in VA in VA Pacbase	Differences at at generation
MP5	Non-numeric line number	Error in the RQ file	Correct in VA Pacbase	Possible error at generation
NEX	Duplicate name of generated entity in a same library	Error in the RQ file	Correct in VA Pacbase: change the external names or select the N variant for programs to be kept with the same model or populate the NR file	To keep several entities that have the same external name, it is required to migrate manually
MEX	Duplicate name of generated MAP in a same library	Error in the RQ file	Correct in VA Pacbase: change the external names	To keep several entities that have the same external name, it is required to migrate manually
NV1	Function level not compatible with the operator	Error in the RQ file	Correct in VA Pacbase: modify the level and/or the operator	Possible generation error
OR1	Orphan conditions	Error in the RQ file	Correct in VA Pacbase	Possible generation error for complex cases

Code	Meaning	Consequence	Possible action	Consequence if no action
RU1	Model entity used in standard entity	Error in the Q file		
PG1	Duplicate P-line with N type	Error in the RQ file	Correct in VA Pacbase	Possible generation error
PG3	Program with a variant or type and structure not yet taken into account	Error in the RQ file	Wait for development	Cannot be migrated
PG4	Program coming from PacReverse not taken into account	Error in the RQ file	Wait for development	Cannot be migrated
SG1	Segment renamed as 00 in a -CD	Error in the RQ file		Generation differences in the migration help
WK	Working line with M C L or A type	Error in the RQ file		Not taken into account
WK1	Working line with a table name that contains a \$ character	Error in the RQ file	Manually correct in RPP or in the MIMA file or after dispatching	Erroneous index of the table
BL	Database Block not to retrieve	Population of NR file	(3)	
DI1	Dialog with an F type (Folder Server) or FV type (Folder view)	Population of NR file	(3)	
EC2	Screen with an F type (Folder Server) or FV type (Folder view)	Population of NR file	(3)	
ET3	Report for PDM layout	Population of NR file	(3)	

Code	Meaning	Consequence	Possible action	Consequence if no action
EU1	Pacdesign User Entity not retrieved	Population of NR file	(3) Caution: this is an option. The input of the procedure must be modified to retrieve it	
EU2	Bulk-type UE not retrieved	Population of NR file	(3)	
MT1	Pacdesign Meta Entity not retrieved	Population of NR file	(3) Caution: this is an option. The input of the procedure must be modified to retrieve it	
MT2	Bulk-type Meta Entity not retrieved	Population of NR file	(3)	
MT3	Data Element of Meta Entity not retrieved	Population of NR file except for a Data Element of 7N 7M 5Q Meta Entities if the MT parameter is positioned in the BVPAMIAM file	(3)	
PA1	Presence of PAF operator	Population of NR file	(3)	
PG2	Program with an obsolete type and structure	Population of NR file	(3)	
UR	Data Element usage not to retrieve	Population of NR file	(3)	
EC4	Screen with only a definition or -CE lines	Population of NS file	(4)	

(1) The CS file of special characters is used to replace characters or strings with valid values.

This file is used by the MIBA, MIBR, MIBJ, and MIMA procedures.

If there is no substitution value, the detected values are set to blank.

(2) The PM file of Programs used as Macros (but that are not typed as Macros) allows the MIBA, MIBR, and MIBJ procedures to process them as Macros.

(3) The NR file contains the entities that are not retrieved after the processing.

This applies to the MIBR and MIBJ procedures.

This file can be edited and populated with obsolete entities.

(4) The NS file contains 'banner' Screens. This file is used later.

MIAM - User Input

Pos.	Len.	Value	Meaning
2	1	'*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3		Extraction Library code
		***	All Libraries
		bbb	bbb Library
22	4	nnnn	Session number (blank = current session)
26	1		Session status
		'T'	If selection of frozen session
		' '	If selection of current or H session

This processing takes into account the parameter file BVPAMIAM, provided at installation with the following structure:

Pos.	Len.	Value	Meaning
1	2	'BL'	Type of block not retrieved
		'CS'	Special character
		'MT'	Forced retrieval of Meta Entity (1)
		'PF'	Indicates whether to retrieve the Programs or Screens that contain PAF operators. By default, they are retrieved. Add a PF line with the value NO in position 3 not to retrieve them.

Pos.	Len.	Value	Meaning
		'UR'	Recognized Data Element usage
		'VO'	Authorized types of COBOL to generate for Screens (2)
		'VP'	Authorized types of COBOL to generate for Programs (2)
		'VS'	Authorized types of COBOL to generate for C/S entities (2)
		'WK'	Type of working line not to retrieve
		'WN'	String not allowed (not compatible with Windows for example)
3	78		Value of the parameter

This file contains the standard parameters provided at installation to allow the procedure to carry out the necessary checks.

This file can be enriched with the wanted special characters. To do so, add a CS line with the wanted character.

(1) The MT parameter is used to force the retrieval of Meta Entities with a code 7N (Endevor), 7M (Environment) or 5Q (Quality) and the retrieval of Pacdesign Meta Entities (use the DS value). Only these four 2-char values are taken into account.

(2) By default, the values of the COBOL types to generate are:

VO: 'NX0130456M7U8FIKQZ'

VP: 'N013458FIKOUXQZC'

VS: 'NX03457OU8FIRQ'.

You can replace them with the values indicated in each parameter. You must indicate all the COBOL types to be recognized in each parameter.

If several lines exist for the same type, only the last line is taken into account.

Note:

It is not necessary to sort the parameter file. Any unknown value is ignored.

Parameters can be removed.

For the Windows platforms, this file is available in the SYS/SKEL directory.

MIAM - Description of Steps

Migration Help Function : BVPLTPAM

Code	Physical name	Type	Label
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7MB	User input	Input	User Input
PAC7CK	Skel dir.: BVPAMIAM	Input	Analysis parameters
PAC7CS	User dir.: MIAMCSPE	Output	Special characters
PAC7NR	User dir.: MIAMNREP	Output	Entities not to retrieve
PAC7NS	User dir.: MIAMSCRN	Output	Screen with only a definition or -CE lines
PAC7PM	User dir.: MIAMPMAC	Output	Programs to be transformed into Macros
PAC7RQ	User dir.: MIAMRQER	Output	Detected errors
PAC7DD	User dir. : MIAMDDPAM	Report	Error report

MIAM - Execution Script

```

| -----
|      VISUALAGE PACBASE
| -----
|      MIGRATION HELP
| -----
|
<job id=MIAM>

<script language="VBScript">
Dim MyProc
MyProc = "MIAM"
</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" , "PLTPAM"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7CK") = BVP_MiamCk
Call BvpEnv("PLTPAM","PAC7CS",Rep_USR & "\MIAMCSPE.txt")
Call BvpEnv("PLTPAM","PAC7NR",Rep_USR & "\MIAMNREP.txt")
Call BvpEnv("PLTPAM","PAC7NS",Rep_USR & "\MIAMSCRN.txt")
Call BvpEnv("PLTPAM","PAC7PM",Rep_USR & "\MIAMPMAC.txt")
Call BvpEnv("PLTPAM","PAC7RQ",Rep_USR & "\MIAMRQER.txt")
Call BvpEnv("PLTPAM","PAC7DD",Rep_USR & "\MIAMDDPM.txt")
Call RunCmdLog ("BVPLTPAM")
Call Err_Cod(Return , 0 , "PLTPAM")

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

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

</script>
</job>

```

MIBA - Batch Migration

MIBA - Introduction

Principle

This procedure is a utility that prepares the batch migration after the extraction of entities in the Repository. It formats files that will be used by the migration tools.

MIBA - User Input

It is the user input of the PACX procedure, with the following specific values:

One '*' line as follows:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code

Pos.	Len.	Value	Meaning
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Extraction Library code
22	4	nnnn	Session number (Blank=current session)
26	1	'T'	Session status if Test session
29	4	EXTR	Extractor code
34	1	'1'	Formatting for UPDP (PAF)
55	1	'1'	Timestamp request indicator

'O'-line: Control cards in front/back of the generation commands

Pos.	Len.	Value	Meaning
2	1	'O'	Line code
3	3	bbb	Generation command
			'GCP': Program
			'GCO': Dialog
			'GGC': Client
			'GGS': Server
7	2	be	Control card in front
9	2	af	Control card in back

You must add the following command line:

Pos.	Len.	Value	Meaning
2	4	'W'	Line code
3	1	'1'	Line number
4	2	'EX'	
6	1	'+'	Library selection code:
			Library and its upper-libraries with the generation of identification lines (* lines)
7	33	Choice	Entity to be extracted, coded in the same way as the 'Choice' field in online mode.
40	4		Extraction type:
		'ALL '	Entity and used entities

Principle of migration

For a given entity, the extraction is performed from each library of each element of its folder, within the same network.

First step: PACX extraction in view '+'

When the definition of an entity is extracted, the Library of each element in its folder is searched for.

The highest-level library is the reference library. The folder of the entity is created later (next step) in each lower-level library found.

In a network, all the libraries that contain an element of the folder are detected.

For each library, a command line in view 'C' of the extracted entity, and an appropriate assign card are formatted and sorted. All these lines make up the input file of the second step.

Second step: PACX extraction in view 'C'

The entities extraction is performed in the extraction library and in the higher-level libraries. An entity folder is created in each extraction library. It is inherited from the higher or equal level libraries.

The PACX extraction has been enhanced:

- When a Segment is extracted, its Data Structure is also extracted.
- When a Screen is extracted, the Definition of its Dialog is also extracted.

Only the erroneous requests are included in the PAC7EZ report.

This procedure requires the PAC7PM file as input to the BVPACS96 program.

This file, defined with 50-character records, contains, in position 1, the VA Pac codes of the Programs used as Macrostructures but not defined as such in the Repository.

This file is produced by the MIAM procedure.

This procedure also allows to replace unknown special characters in an entity code with other characters or to replace unauthorized words (Windows reserved words for example) with other codes. The matching table is the PAC7CS file used as input to the BVPACS94 and BVPACS96 programs (optional file, created by the MIAM procedure).

If it exists, this file must be specified in the BVPMIBA script, by filling in the WshEnv("BVP_MibaSpecChar") line.

If the line is not valued, a MibaSpecChar.txt file will be created, empty, in the [database_name] directory.

This procedure also generates the input transactions for the procedures:

- MIMA (migration of macrostructures): GCM and GMO commands.
- GPRT (PRINT - GENERATION): GCP, GCO, GGC and GGS commands
- MIA1 (migration of Programs): GCP commands
- MIA2 (migration of Screens): GCO commands
- MIA3 (migration of Clients): GGC commands
- MIA4 (migration of Servers): GGS commands
- The control cards in front/back of these records come from the 'O'-lines of user input.

.Notes: GMO commands: creation of high priority Macros

All the specific lines of a Screen that cause the move of a Macro function or subfunction are gathered in the high priority Macro.

When the repository is migrated, the Screens that contain specific relative positions (*A, *P, *C, *R) are detected. A GMO command line is created for each of these Screens, which will be transformed into high priority Macros.

These Macros will be generated by the MIMA procedure under the name: Screen code suffixed by 'SP'.

MIBA - Description of Steps

Extraction: PACX

This step extracts transactions according to user input.

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

Code	Physical name	Type	Label
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7MB	User Input	Input	User input
PAC7MA	NUL	Input	Work file
PAC7ES	NUL	Input	Work file
PAC7BM	Tmp dir.: WBM	Input/Output	User input
PAC7MM	Tmp dir.: WMM	Input/Output	Work file
PAC7MJ	Tmp dir.: WMJ	Input/Output	Work file
PAC7TE	Tmp dir.: WTE	Input/Output	Work file
PAC7RE	Tmp dir.: WRE	Input/Output	Work file
PAC7RM	Tmp dir.: WRM	Input/Output	Work file
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted transactions
SYSEXT	Tmp dir.: WSY	Input/Output	Work file
PAC7MV	Tmp dir.: WMV	Output	Work file
PAC7CP	NULCP	Output	Work file
PAC7MR	Tmp dir.: WMR	Output	Work file
PAC7MX	Tmp dir.: WMX	Output	Entities not extracted (PACX)
PAC7GY	NUL	Output	Extracted transactions for UPDP
PAC7RQ	Tmp dir.: WQR	Output	Matching Entity/Library
PAC7TD	Tmp dir.: WTD	Output	Work file
PAC7UE	Tmp dir.: WUE	Output	Work file
PAC7IA	User dir.: PACXIA	Report	General printout of the program stream
PAC7DD	User dir.: PACXDD	Report	Errors on input transactions
PAC7ED	Tmp dir.: WED	Report	Extractions report
PAC7EE	User dir.: PACXEE	Report	Extractions report
PAC7EG	Tmp dir.: WEG	Report	Extractions report
PAC7EM	Tmp dir.: WEM	Report	Extractions report
PAC7EP	User dir.: PACXEP	Report	Extractions report
PAC7EQ	User dir.: PACXEQ	Report	Extractions report
PAC7EU	Tmp dir.: WEU	Report	Extractions report

Code	Physical name	Type	Label
PAC7EZ	User dir.: PACXEZ	Report	Extractions report

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD)

Sort and formatting of command lines: BVPACS92

Code	Physical name	Type	Label
PAC7QR	Tmp dir.: WQR	Input	Matching Entity/Library
PAC7RQ	Tmp dir.: WPACXRQ	Output	Matching Entity/Library after sort
PAC7MZ	Tmp dir.: WMB	Output	Command lines for PACX input

Extraction: PACX2

This step extracts the transactions according to the user input generated by the BVPACS92 program.

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AN	Database dir. : AN	Input	Development Database index
PAC7AR	Database dir. : AR	Input	Development Database data
PAC7AY	Database dir. : AY	Input	Development Database extension data
PACGGN	Admin Database - DBase dir. : AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7MB	Tmp dir. : WMB	Input	User input
PAC7MA	NUL	Input	Work file
PAC7ES	NUL	Input	Work file
PAC7BM	Tmp dir. : WBM	Input/Output	User input

Code	Physical name	Type	Label
PAC7MM	Tmp dir. : WMM	Input/Output	Work file
PAC7MJ	Tmp dir. : WMJ	Input/Output	Work file
PAC7TE	Tmp dir. : WTE	Input/Output	Work file
PAC7RE	Tmp dir. : WRE	Input/Output	Work file
PAC7RM	Tmp dir. : WRM	Input/Output	Work file
PAC7WD	Tmp dir. : WWD	Input/Output	Extracted transactions
SYSEXT	Tmp dir. : WSY2	Input/Output	Work file
PAC7MV	Tmp dir. : WMV	Output	Work file
PAC7MR	Tmp dir. : WMR	Output	Work file
PAC7MX	Tmp dir. : WMX	Output	Entities not extracted (PACX)
PAC7CP	Tmp dir. : WCP	Output	Work file
PAC7GY	Tmp dir. : WGY	Output	Extracted transactions for UPDP
PAC7RQ	NUL	Output	Matching Entity/Library
PAC7TD	Tmp dir. : WTD	Output	Work file
PAC7UE	Tmp dir. : WUE	Output	Work file
PAC7IA	User dir. : PACX2IA	Report	General printout of the program stream
PAC7DD	User dir. : PACX2DD	Report	Errors on input transactions
PAC7ED	Tmp dir. : WED	Report	Extractions report
PAC7EE	User dir. : PACX2EE	Report	Extractions report
PAC7EG	Tmp dir. : WEG	Report	Extractions report
PAC7EM	Tmp dir. : WEM	Report	Extractions report
PAC7EP	User dir. : PACX2EP	Report	Extractions report
PAC7EQ	User dir. : PACX2EQ	Report	Extractions report
PAC7EU	Tmp dir. : WEU	Report	Extractions report
PAC7EZ	User dir. : PACX2EZ	Report	Extractions report

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD)

Batch migration: BVPACS94

Code	Physical name	Type	Label
PAC7RQ	Tmp dir.: WPACXRQ	Input	Matching Entity/lowest Library
PAC7CS	User dir. : MibaSpecChar	Input	Special characters matching file
PAC7QR	Tmp dir.: WPAC7QR	Output	Matching Entity/lowest Library

Batch migration : BVPACS96

Code	Physical name	Type	Label
PAC7MB	User input	Input	User input
PAC7GY	Tmp dir.: WGY	Input	File resulting from PACX extraction
PAC7PM	User dir.: MvMiba	Input	File with the list of Pgms to be transformed into PMS
PAC7CS	User dir.: MibaSpecChar	Input	Special characters matching file
PAC7TA	Tmp dir.: WTA	Output	File containing transactions grouped by type
PAC7MM	User dir.: INPUTMIMA	Output	File containing MIMA commands
PAC7MI	User dir.: MIA1	Output	File containing GCP commands
PAC7MO	User dir.: MIA2	Output	File containing GCO commands
PAC7MC	User dir.: MIA3	Output	File containing GGC commands
PAC7MS	User dir.: MIA4	Output	File containing GGS commands

Reorganization of transactions : BVPACS97

Code	Physical name	Type	Label
PAC7TA	Tmp dir.: WTA	Input	File containing transactions grouped by type
PAC7AT	Tmp dir.: WAT	Output	File with transactions grouped by type, sorted again

Sort and formatting of import file: BVPACS91

Code	Physical name	Type	Label
PAC7AT	Tmp dir.: WAT	Input	File containing transactions grouped by type
PAC7TI	Tmp dir.: WTI	Output	File with transactions grouped by type, sorted again

Global sort: BVPACS98

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database data
PAC7AE	System - Skel dir.: AE	Input	Error messages
PAC7CP	Tmp dir.: WCP	Input	Counter of extracted entities
PAC7RQ	Tmp dir.: WPAC7QR	Input	Matching Entity/lowest Library
PAC7TI	Tmp dir.: WTI	Input	File with transactions grouped by type, sorted again
PAC7TG	User dir.: MIBA	Output	Result file with global sort

MIBA - Execution Script

```

'-----
'      VISUALAGE PACBASE
'-----
'      BATCH MIGRATION
'-----
'
<job id=MIBA>

<script language="VBScript">
Dim MyProc
MyProc = "MIBA"
</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" , "PACX"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGR") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACX", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACX", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("PACX", "PAC7MM", Rep_TMP & "\WMM.tmp")
Call BvpEnv("PACX", "PAC7MJ", Rep_TMP & "\WMJ.tmp")
Call BvpEnv("PACX", "PAC7TE", Rep_TMP & "\WTE.tmp")

```

```

Call BvpEnv("PACX","PAC7RE",Rep_TMP & "\\WRE.tmp")
Call BvpEnv("PACX","PAC7RQ",Rep_TMP & "\\WQR.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\\NULCP.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\\PACXEZ.txt")
Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS92"))
'-----
Call BvpEnv("PACS92","PAC7QR",Rep_TMP & "\\WQR.tmp")
Call BvpEnv("PACS92","PAC7RQ",Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS92","PAC7MZ",Rep_TMP & "\\WMB.tmp")
Call RunCmdLog ("BVPACS92")
Call Err_Cod(Return , 0 , "PACS92")

Call Msg_Log (Array("1022" , "PACX"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
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"
Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\\WMB.tmp")
Call BvpEnv("PACX","PAC7BM",Rep_TMP & "\\WBM.tmp")
Call BvpEnv("PACX","PAC7WD",Rep_TMP & "\\WWD.tmp")
Call BvpEnv("PACX","PAC7MM",Rep_TMP & "\\WMM.tmp")
Call BvpEnv("PACX","PAC7MJ",Rep_TMP & "\\WMJ.tmp")
Call BvpEnv("PACX","PAC7TE",Rep_TMP & "\\WTE.tmp")

```

```

Call BvpEnv("PACX","PAC7RE",Rep_TMP & "\WRE.tmp")
Call BvpEnv("PACX","PAC7RQ",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\WCP.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\PACX2IA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\PACX2DD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\PACX2EE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\PACX2EP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\PACX2EQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\WEU.tmp")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\PACX2EZ.txt")
Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\WSY2.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS94"))
'-----
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACS94","PAC7RQ",Rep_TMP & "\WPACXRQ.tmp")
Call BvpEnv("PACS94","PAC7QR",Rep_TMP & "\WPAC7QR.tmp")
Call RunCmdLog ("BVPACS94")
Call Err_Cod(Return , 0 , "PACS94")

Call Msg_Log (Array("1022" , "PACS96"))
'-----
Call BvpEnv("PACS96","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
WshEnv("PAC7PM") = BVP_MibaFi
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACS96","PAC7TA",Rep_TMP & "\WTA.tmp")
Call BvpEnv("PACS96","PAC7MM",Rep_USR & "\INPUTMIMA.txt")
Call BvpEnv("PACS96","PAC7MI",Rep_USR & "\MIA1.txt")
Call BvpEnv("PACS96","PAC7MO",Rep_USR & "\MIA2.txt")
Call BvpEnv("PACS96","PAC7MC",Rep_USR & "\MIA3.txt")
Call BvpEnv("PACS96","PAC7MS",Rep_USR & "\MIA4.txt")
Call RunCmdLog ("BVPACS96")
Call Err_Cod(Return , 0 , "PACS96")

```

```

Call Msg_Log (Array("1022" , "PACS97"))
'-----
Call BvpEnv("PACS97","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACS97","PAC7TA",Rep_TMP & "\WTA.tmp")
Call RunCmdLog ("BVPACS97")
Call Err_Cod(Return , 0 , "PACS97")

Call Msg_Log (Array("1022" , "PACS91"))
'-----
Call BvpEnv("PACS91","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACS91","PAC7TI",Rep_TMP & "\WTI.tmp")
Call RunCmdLog ("BVPACS91")
Call Err_Cod(Return , 0 , "PACS91")

Call Msg_Log (Array("1022" , "PACS98"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PACS98","PAC7CP",Rep_TMP & "\WCP.tmp")
Call BvpEnv("PACS98","PAC7TI",Rep_TMP & "\WTI.tmp")
Call BvpEnv("PACS98","PAC7RQ",Rep_TMP & "\WPAC7QR.tmp")
Call BvpEnv("PACS98","PAC7TG",Rep_USR & "\MIBA.ta2")
Call RunCmdLog ("BVPACS98")
Call Err_Cod(Return , 0 , "PACS98")

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

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

</script>
</job>

```

MIBR - Batch Migration of the Context

MIBR - Introduction

Principle

This procedure is a utility that prepares the batch migration after the extraction of entities from the Repository in a given context (session and ascending or descending network of a library). It formats files that will be used by the migration tools.

MIBR - User Input

Pos.	Len.	Value	Meaning
2	1	">*	Line code
3	8	uuuuuuuu	User code

Pos.	Len.	Value	Meaning
11	8	pppppppp	User password
22	4	nnnn	Session number (Blank = current session)
26	1		Session status
		'T'	If selection of frozen session
		' '	If current or H session
33	1		Code of library extraction
		'A'	Extraction of a lower level library and its higher level libraries
		'D'	Extraction of a library and its depending libraries
		'U'	Extraction of only one library

As many lines as Libraries to be extracted.

Pos.	Len.	Value	Meaning
3	3	bbb	Code of the library to be extracted

'O'-line: Control cards in front/back of generation commands

Pos.	Len.	Value	Meaning
2	1	'O'	Line code
3	3	bbb	Generation command
			'GCP': Program
			'GCO': Dialog
			'GGC': Client
			'GGS': Server
7	2	be	Control card in front
9	2	af	Control card in back

'P' line: project created after the libraries split

Pos.	Len.	Value	Meaning
2	1	'P'	Line code
3	3	bbb	Code of library to split
6	3	ccc	Type of Pacbase entity to move
			'E ': Data Element

Pos.	Len.	Value	Meaning
			'S ': Data Structure and Segment
			'R ': Report
			'B ': Database Block
			'P ': Program
			'O ': Dialog and Screen
			'F ': Client Meta Entity
			'\$tt': Client User Entity (tt = call type)
			'T ': Text
9	15		Code of the new project

Principle of migration

An extraction context is selected. The session and the library, from which the ascending or descending network will be extracted, are specified.

First step: Extraction of entities present in the context.

All the indexes in the database are read sequentially.

The selection is made on the primary indexes of all the entity families. The highest level library of the entity definition is the reference library. If an element of the folder is detected in a lower level library, the entity folder will be created later in this library (next step).

The overrides of the entity definitions and complete folders are thus detected in a libraries network (it means the multiple presence of the definition or the existence of one or more elements of its folder in a library different from the definition library).

In a network, all the libraries that contain an element of the folder are detected.

For each library, a command line in view 'C' of the extracted entity, with an appropriate assign card, is formatted and sorted. All these lines make up the input file of the third step.

Second step: consideration of entities not retrieved

This step allows not to retrieve in output all the entities listed in the PAC7NR file.

The PAC7NR file of the entities not to be retrieved (file produced by the MIAM procedure) must exist.

If it exists, this file must be specified in the execution script, by filling in the WshEnv("BVP_MibrNotRet") line.

If the line is not valued, an Mvmibrnr.txt file will be created, empty, in the [database_name] directory.

Third step: PACX extraction in view 'C'

The entities are extracted in the extraction library and in its higher-level libraries. An entity folder is created in each extraction library. It is inherited from the higher or equal level libraries.

The PACX extraction has been enhanced:

- When a Segment is extracted, its Data Structure is also extracted.
- When a Screen is extracted, the Definition of its Dialog is also extracted.

Erroneous requests only are printed in the PAC7EZ report.

This procedure requires the PAC7PM file as input to the BVPACS96 program.

This file, defined with 50-character records, contains in position 1 the VA Pacbase codes of the Programs used as Macrostructures but not defined as such in the Repository.

This file is produced by the MIAM procedure.

This procedure also allows to replace unknown special characters in an entity code with another character, or to replace Windows reserved words with another code.

The matching table is represented by the PAC7CS file used as input to the BVPACS94 and BVPACS96 programs (optional file, created by the MIAM procedure).

This procedure also generates the input transactions for the following procedures:

- MIMA (migration of macrostructures): GCM and GMO commands.
- GPRT (PPRINT - GENERATION): GCP, GCO, GGC and GGS commands
- MIA1 (migration of Programs): GCP commands
- MIA2 (migration of Screens): GCO commands
- MIA3 (migration of Clients): GGC commands

- MIA4 (migration of Servers): GGS commands
- The control cards in front/back of these records come from the 'O'-lines of user input. (See subchapter "MIAx - Optional control cards")

.Notes: GMO commands: creation of high priority Macros

All the specific lines of a Screen that cause the move of a Macro function or subfunction are gathered in the high priority Macro.

When the repository is migrated, the Screens that contain specific relative positions (*A, *P, *C, *R) are detected. A GMO command line is created for each of these Screens, which will be transformed into high priority Macros.

These Macros will be generated by the MIMA procedure under the name: Screen code suffixed by 'SP'.

MIBR - Split of Libraries

Some very large libraries can be reduced by the MIBR procedure because this procedure splits them.

The library to be reduced is divided into N new projects.

Using P lines, you must specify the new project where each relevant Pacbase entity will be moved to. All the projects are dependent and their hierarchy depends on the associated Pacbase entity type.

The order is described in the input table of the P line.

However, you can put the projects of the User Entities wherever you want to.

The same project can be associated with several Pacbase entity types if their types are directly dependent (immediate hierarchy).

To ensure that the list of the new projects created by the library split is coherent, a control verifies that all the Pacbase entity types (except the Meta Entities) are present in the P lines.

Limits of P input:

- 5 libraries to split
- 40 projects by library

It means 30 projects for the User Entities and one project by type for all the other Pacbase entities. The generic code \$** can be used to group all the User

Entities in the same project. You can create one project for one given User Entity and another project for all the other UE (\$**).

- The project code cannot contain the special characters ' \ /:%?<>"| ' or be the reserved word 'INTER'.

Rules of use:

To use this option, the migration request is made in descending mode 'D'.

Order of MIBR user input:

- All the cards of libraries networks extraction.
- All the optional cards in front/back.
- All the projects cards.

Respect the hierarchy of the libraries to split by sorting the P lines from the highest to the lowest level libraries.

The MIBR procedure saves the dispatching user requests. Thus, they can be taken into account by the MIBJ procedure.

The SPLITLIB file produced by MIBR will be used as input to the MIBJ procedure.

It contains all the projects coming from the libraries split and their hierarchy.

If it exists, this file must be specified in the execution script, by filling in the WshEnv("BVP_MibrSplitLib") line.

If the line is not valued, a SPLITLIB.txt file will be created, empty, in the [database_name] directory.

MIBR - Description of Steps

Loading the NR file: entities not retrieved: BVPACS9B

Code	Physical name	Type	Label
PAC7RN	User dir.: MibrNotRet	Input	Entities not retrieved, produced by MIAM
PAC7NR	Tmp dir. : WNR	Output	Entities not retrieved, produced by MIAM, formatted

Network extraction: BVPACS21

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGN	Admin Database - DBase dir. : AN	Input	Administration Database index
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir. : AN	Input	Development Database index
PAC7AY	Database dir. : AY	Input	Development Database extension data
PAC7MB	User input	Input	User transactions
PAC7RQ	Tmp dir. : WQR	Output	Matching Entity/lowest Library
PAC7DD	User dir. : MIBRDDS21	Report	Error report

Processing of entities not retrieved: BVPACS9A

Code	Physical name	Type	Label
PAC7NR	Tmp dir.: WNR	Input	Entities not retrieved, produced by MIAM
PAC7QR	Tmp dir.: WQR	Input	Matching Entity / lowest level library
PAC7RQ	Tmp dir.: WRQ	Output	Matching Entity / refined library

Sort and formatting of command lines: BVPACS92

Code	Physical name	Type	Label
PAC7QR	Tmp dir.: WRQ	Input	Matching Entity/Library
PAC7RQ	Tmp dir.: WPACXRQ	Output	Matching Entity/Library after sort
PAC7MZ	Tmp dir.: WMB	Output	Command lines for PACX input

Extraction: PACX

This step extracts the transactions according to the user input generated by the BVPACS92 program.

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AN	Database dir. : AN	Input	Development Database index
PAC7AR	Database dir. : AR	Input	Development Database data
PAC7AY	Database dir. : AY	Input	Development Database extension data
PACGGN	Admin Database - DBase dir. : AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7MB	User input	Input	User input
PAC7MA	NUL	Input	Work file
PAC7ES	NUL	Input	Work file
PAC7BM	Tmp dir.: WBM	Input/Output	User input
PAC7MM	Tmp dir.: WMM	Input/Output	Work file
PAC7MJ	Tmp dir.: WMJ	Input/Output	Work file
PAC7TE	Tmp dir.: WTE	Input/Output	Work file
PAC7RE	Tmp dir.: WRE	Input/Output	Work file
PAC7RM	Tmp dir.: WRM	Input/Output	Work file
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted transactions
SYSEXT	Tmp dir.: WSY	Input/Output	Work file
PAC7MV	Tmp dir.: WMV	Output	Work file
PAC7MR	Tmp dir.: WMR	Output	Work file
PAC7MX	Tmp dir.: WMX	Output	Non extracted entities (PACX)
PAC7GY	Tmp dir.: WGY	Output	Extracted transactions for UPDP
PAC7RQ	NUL	Output	Matching Entity/Library
PAC7TD	Tmp dir.: WTD	Output	Work file
PAC7UE	Tmp dir.: WUE	Output	Work file
PAC7CP	Tmp dir. : WCP	Output	Work file
PAC7IA	User dir.: PACXIA	Report	General printout of the program stream
PAC7DD	User dir.: PACXDD	Report	Errors on input transactions

Code	Physical name	Type	Label
PAC7ED	Tmp dir.: WED	Report	Extractions report
PAC7EE	User dir.: PACXEE	Report	Extractions report
PAC7EG	Tmp dir.: WEG	Report	Extractions report
PAC7EM	Tmp dir.: WEM	Report	Extractions report
PAC7EP	User dir.: PACXEP	Report	Extractions report
PAC7EQ	User dir.: PACXEQ	Report	Extractions report
PAC7EU	Tmp dir.: WEU	Report	Extractions report
PAC7EZ	User dir.: PACXEZ	Report	Extractions report

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD)

Batch migration: BVPACS94

Code	Physical name	Type	Label
PAC7RQ	Tmp dir.: WPACXRQ	Input	Matching Entity/lowest Library
PAC7CS	User dir.: MibaSpecChar	Input	Special characters matching file
PAC7QR	User dir.: SPLITLIB	Output	Matching Entity/lowest Library

Batch migration : BVPACS96

Code	Physical name	Type	Label
PAC7MB	User input	Input	User input
PAC7GY	Tmp dir.: WGY	Input	File resulting from PACX extraction
PAC7PM	User dir.: MvMiba	Input	File with the list of Pgms to be transformed into PMS
PAC7CS	User dir.: MibaSpecChar	Input	Special characters matching file
PAC7TA	Tmp dir.: WTA	Output	File containing transactions grouped by type
PAC7MM	User dir.: INPUTMIMA	Output	File containing MIMA commands
PAC7MI	User dir.: MIA1	Output	File containing GCP commands
PAC7MO	User dir.: MIA2	Output	File containing GCO commands

Code	Physical name	Type	Label
PAC7MC	User dir.: MIA3	Output	File containing GGC commands
PAC7MS	User dir.: MIA4	Output	File containing GGS commands

Reorganization of transactions : BVPACS97

Code	Physical name	Type	Label
PAC7TA	Tmp dir.: WTA	Input	File containing transactions grouped by type
PAC7AT	Tmp dir.: WAT	Output	File with transactions grouped by type, sorted again

Sort and formatting of import file: BVPACS91

Code	Physical name	Type	Label
PAC7AT	Tmp dir.: WAT	Input	File containing transactions grouped by type
PAC7TI	Tmp dir.: WTI	Output	File with transactions grouped by type, sorted again

Global sort : BVPACS98

Code	Physical name	Type	Label
PAC7AR	Database dir. : AR	Output	Development Database data
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7CP	Tmp dir. : WCP	Input	Counter of extracted entities
PAC7RQ	User dir.: SPLITLIB	Input	Matching Entity/Lowest level Library
PAC7TI	Tmp dir. : WAT	Input	File containing the sorted transactions
PAC7TG	User dir. : MIBR	Output	Result file of global sort

MIBR - Execution Script

```

|-----|
| VISUALAGE PACBASE |
|-----|
| BATCH MIGRATION |
|-----|
|

```

```

<job id=MIBR>

<script language="VBScript">
Dim MyProc
MyProc = "MIBR"
</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" , "PACS9B"))
'-----
WshEnv("PAC7RN") = BVP_MibrNr
Call BvpEnv("PACS9B","PAC7NR",Rep_TMP & "\\WNR.tmp")
Call RunCmdLog ("BVPACS9B")
Call Err_Cod(Return , 0 , "PACS9B")

Call Msg_Log (Array("1022" , "PACS21"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
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("PAC7MB") = Fic_Input
Call BvpEnv("PACS21","PAC7DD",Rep_USR & "\\MIBRDDS21.txt")
Call BvpEnv("PACS21","PAC7RQ",Rep_TMP & "\\WQR.tmp")
Call RunCmdLog ("BVPACS21")
Call Err_Cod(Return , 0 , "PACS21")

Call Msg_Log (Array("1022" , "PACS9A"))
'-----
Call BvpEnv("PACS9A","PAC7NR",Rep_TMP & "\\WNR.tmp")
Call BvpEnv("PACS9A","PAC7QR",Rep_TMP & "\\WQR.tmp")
Call BvpEnv("PACS9A","PAC7RQ",Rep_TMP & "\\WRQ.tmp")
Call RunCmdLog ("BVPACS9A")
Call Err_Cod(Return , 0 , "PACS9A")

Call Msg_Log (Array("1022" , "PACS92"))
'-----
Call BvpEnv("PACS92","PAC7QR",Rep_TMP & "\\WRQ.tmp")
Call BvpEnv("PACS92","PAC7RQ",Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS92","PAC7MZ",Rep_TMP & "\\WMB.tmp")
Call RunCmdLog ("BVPACS92")
Call Err_Cod(Return , 0 , "PACS92")

Call Msg_Log (Array("1022" , "PACX"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
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"
Call BvpEnv("PACX", "PAC7MB", Rep_TMP & "\WMB.tmp")
Call BvpEnv("PACX", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACX", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("PACX", "PAC7MM", Rep_TMP & "\WMM.tmp")
Call BvpEnv("PACX", "PAC7MJ", Rep_TMP & "\WMJ.tmp")
Call BvpEnv("PACX", "PAC7TE", Rep_TMP & "\WTE.tmp")
Call BvpEnv("PACX", "PAC7RE", Rep_TMP & "\WRE.tmp")
Call BvpEnv("PACX", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7RM", Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX", "PAC7MA", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7ES", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7UE", Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX", "PAC7CP", Rep_TMP & "\WCP.tmp")
Call BvpEnv("PACX", "PAC7TD", Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX", "PAC7IA", Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX", "PAC7DD", Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX", "PAC7ED", Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX", "PAC7EE", Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX", "PAC7EG", Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX", "PAC7EM", Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX", "PAC7EP", Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX", "PAC7EQ", Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX", "PAC7EU", Rep_TMP & "\WEU.tmp")
Call BvpEnv("PACX", "PAC7EZ", Rep_USR & "\PACXEZ.txt")
Call BvpEnv("PACX", "SYSEXT", Rep_TMP & "\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS94"))
'-----
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACS94", "PAC7RQ", Rep_TMP & "\WPACXRQ.tmp")
Call BvpEnv("PACS94", "PAC7QR", Rep_USR & "\SPLITLIB.txt")
Call RunCmdLog ("BVPACS94")
Call Err_Cod(Return , 0 , "PACS94")

Call Msg_Log (Array("1022" , "PACS96"))
'-----
Call BvpEnv("PACS96", "PAC7GY", Rep_TMP & "\WGY.tmp")
WshEnv("PAC7CS") = BVP_MibaCs

```

```

WshEnv("PAC7PM") = BVP_MibaFi
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACS96","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call BvpEnv("PACS96","PAC7MM",Rep_USR & "\\INPUTMIMA.txt")
Call BvpEnv("PACS96","PAC7MI",Rep_USR & "\\MIA1.txt")
Call BvpEnv("PACS96","PAC7MO",Rep_USR & "\\MIA2.txt")
Call BvpEnv("PACS96","PAC7MC",Rep_USR & "\\MIA3.txt")
Call BvpEnv("PACS96","PAC7MS",Rep_USR & "\\MIA4.txt")
Call RunCmdLog ("BVPACS96")
Call Err_Cod(Return , 0 , "PACS96")

Call Msg_Log (Array("1022" , "PACS97"))
'-----
Call BvpEnv("PACS97","PAC7AT",Rep_TMP & "\\WAT.tmp")
Call BvpEnv("PACS97","PAC7TA",Rep_TMP & "\\WTA.tmp")
Call RunCmdLog ("BVPACS97")
Call Err_Cod(Return , 0 , "PACS97")

Call Msg_Log (Array("1022" , "PACS91"))
'-----
Call BvpEnv("PACS91","PAC7AT",Rep_TMP & "\\WAT.tmp")
Call BvpEnv("PACS91","PAC7TI",Rep_TMP & "\\WTI.tmp")
Call RunCmdLog ("BVPACS91")
Call Err_Cod(Return , 0 , "PACS91")

Call Msg_Log (Array("1022" , "PACS98"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
WshEnv("PAC7AR") = Rep_BASE & "\\AR"
Call BvpEnv("PACS98","PAC7CP",Rep_TMP & "\\WCP.tmp")
Call BvpEnv("PACS98","PAC7TI",Rep_TMP & "\\WTI.tmp")
Call BvpEnv("PACS98","PAC7RQ",Rep_USR & "\\SPLITLIB.txt")
Call BvpEnv("PACS98","PAC7TG",Rep_USR & "\\MIBR.ta2")
Call RunCmdLog ("BVPACS98")
Call Err_Cod(Return , 0 , "PACS98")

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

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

</script>
</job>

```

MIBJ - Migration of the journal

MIBJ - Introduction

Principle

The procedure for the journal transactions migration (MIBJ) applies to the archived journal file of the repository. It formats files used by the migration tools.

- It makes it possible to select transactions in the journal in a range of dates, sessions, libraries, etc.
- It makes it possible to extract, from the repository, the entities of the selected transactions.

The extraction is performed in the library and session of the transaction, and in all its lower-level libraries where overrides of the definition or of an element in the entity folder are detected.

For each entity, the highest-level library in the repository where the entity is detected (source library) is indicated.

The created or modified entities are extracted, the deleted entities are listed.

MIBJ - User input

A '*' line with user code and password.

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

A 'J' line with the extraction criteria :

Pos.	Len.	Value	Meaning
2	1	'J'	Line code
24	4	ssss	Beginning session number
28	4	ssss	End session number
32	8	ccyymmdd	Beginning date
40	8	ccyymmdd	End date
48	1		Version of the selected transactions
		' '	All sessions
		'Z'	Current session only
		'T'	Frozen session only
49	3	bbb	Code of the selected library

'O'-line: Control cards in front/back of the generation commands

Pos.	Len.	Value	Meaning
2	1	'O'	Line code
3	3	bbb	Generation command
			'GCP': Program
			'GCO': Dialog
			'GGC': Client
			'GGS': Server
7	2	be	Control card in front
9	2	af	Control card in back

Principle of migration

For each journal transaction, definition or entity folder, the extraction is performed from the library and session of the transaction and from each library of each element of its folder, in the lower network.

First step: journal analysis

For each transaction, definition or folder, an RQ record is formatted with the entity type, entity code, the transaction library and session.

When a user entity is processed, its meta entity is extracted.

A deleted entity is noted in the RQ file.

The list of frozen sessions is specified in the RQ file.

For each entity, the procedure searches in the repository for its highest-level library and all the libraries higher and lower than the journal transaction library where the entity is overridden (the multiple presence of the definition or the existence of one or more parts of its folder in a library different from the definition library).

The highest-level library is the reference library. In each lower library found, the folder of the entity will be created (next step).

For each library, a command line in 'C' view of the extracted entity, and an appropriate assign card are formatted and sorted.

All these lines make up the input file of the third step.

Second step: take into account the entities not retrieved

This step allows not to transfer in output all the entities that are listed in the PAC7NR file.

This step requires the presence of the PAC7NR file (entities not retrieved) produced by the MIAM procedure.

Third step: PACX extraction in 'C' view

The entities are extracted in the extraction library and in the higher-level libraries.

In each extraction library, an entity folder is created, inherited from its higher or equal level libraries.

Erroneous requests only are printed in the PAC7EZ report.

Split of libraries

The MIBR procedure saves the user requests of libraries split. They are taken into account in the MIBJ procedure. The SPLITLIB file, produced by MIBR, is used as input to MIBJ. It contains all the projects created by the libraries split and their hierarchy. Each entity extracted by MIBJ and present in the PAC7RQ file is processed to verify whether it must be moved to a new project, i.e. if its target library is a library to split. If so, the new project specific to its entity type is assigned to this entity.

MIBJ - Description of steps

Loading the NR file: entities not retrieved: BVPACS9B

Code	Physical name	Type	Label
PAC7RN	User dir.: MibrNotRet	Input	Entities not retrieved, produced by MIAM
PAC7NR	Tmp dir. : WNR	Output	Entities not retrieved, produced by MIAM, formatted

Journal extraction : BVPACSJR

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

Code	Physical name	Type	Label
PACGGU	Admin Database - Base dir. : GU	Input	Administration Database users
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
PAC7MB	User Input	Input	User transactions
PAC7PJ	Save dir. : PJ	Output	Archived transactions
PAC7RQ	Tmp dir. : WRQJR	Output	Matching Entity/lowest Library
PAC7DD	User dir. : MIBJDDSJR	Report	Error report
PAC7EE	User dir. : MIBJEESJR	Report	Extractions report

override detection : BVPACS93

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Base dir. : AR	Input	Development Database data
PAC7AN	Base dir. : AN	Input	Development Database index
PAC7SL	Tmp dir.: SPLITLIB	Input	Splitting Library
PAC7RQ	Tmp dir. : WRQJR	Input	Matching Entity/Library
PAC7QR	Tmp dir. : WQR	Output	Matching Entity/Library after sort

Processing of entities not retrieved: BVPACS9A

Code	Physical name	Type	Label
PAC7NR	Tmp dir.: WNR	Input	Entities not retrieved, produced by MIAM
PAC7QR	Tmp dir.: WQR	Input	Matching Entity / lowest level library
PAC7RQ	Tmp dir.: WRQ	Output	Matching Entity / refined library

Sort and formatting of command lines: BVPACS92

Code	Physical name	Type	Label
PAC7QR	Tmp dir.: WRQ	Input	Matching Entity/Library
PAC7RQ	Tmp dir.: WPACXRQ	Output	Matching Entity/Library after sort

Code	Physical name	Type	Label
PAC7MZ	Tmp dir.: WMB	Output	Command lines for PACX input

Extraction: PACX

This step extracts the transactions according to the user input generated by the BVPACS92 program.

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AN	Database dir. : AN	Input	Development Database index
PAC7AR	Database dir. : AR	Input	Development Database data
PAC7AY	Database dir. : AY	Input	Development Database extension data
PACGGN	Admin Database - DBase dir. : AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7MB	User input	Input	User input
PAC7MA	NUL	Input	Work file
PAC7ES	NUL	Input	Work file
PAC7BM	Tmp dir.: WBM	Input/Output	User input
PAC7MM	Tmp dir.: WMM	Input/Output	Work file
PAC7MJ	Tmp dir.: WMJ	Input/Output	Work file
PAC7TE	Tmp dir.: WTE	Input/Output	Work file
PAC7RE	Tmp dir.: WRE	Input/Output	Work file
PAC7RM	Tmp dir.: WRM	Input/Output	Work file
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted transactions
SYSEXT	Tmp dir.: WSY	Input/Output	Work file
PAC7MV	Tmp dir.: WMV	Output	Work file
PAC7MR	Tmp dir.: WMR	Output	Work file
PAC7MX	Tmp dir.: WMX	Output	Non extracted entities (PACX)
PAC7GY	Tmp dir.: WGY	Output	Extracted transactions for UPDP

Code	Physical name	Type	Label
PAC7RQ	NUL	Output	Matching Entity/Library
PAC7TD	Tmp dir.: WTD	Output	Work file
PAC7UE	Tmp dir.: WUE	Output	Work file
PAC7CP	Tmp dir. : WCP	Output	Work file
PAC7IA	User dir.: PACXIA	Report	General printout of the program stream
PAC7DD	User dir.: PACXDD	Report	Errors on input transactions
PAC7ED	Tmp dir.: WED	Report	Extractions report
PAC7EE	User dir.: PACXEE	Report	Extractions report
PAC7EG	Tmp dir.: WEG	Report	Extractions report
PAC7EM	Tmp dir.: WEM	Report	Extractions report
PAC7EP	User dir.: PACXEP	Report	Extractions report
PAC7EQ	User dir.: PACXEQ	Report	Extractions report
PAC7EU	Tmp dir.: WEU	Report	Extractions report
PAC7EZ	User dir.: PACXEZ	Report	Extractions report

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD)

Batch migration: BVPACS94

Code	Physical name	Type	Label
PAC7RQ	Tmp dir.: WPACXRQ	Input	Matching Entity/lowest Library
PAC7CS	User dir. : MibaSpecChar	Input	Special characters matching file
PAC7QR	Tmp dir.: WPAC7QR	Output	Matching Entity/lowest Library

Batch migration : BVPACS96

Code	Physical name	Type	Label
PAC7MB	User input	Input	User input
PAC7GY	Tmp dir.: WGY	Input	File resulting from PACX extraction

Code	Physical name	Type	Label
PAC7PM	User dir.: MvMiba	Input	File with the list of Pgms to be transformed into PMS
PAC7CS	User dir.: MibaSpecChar	Input	Special characters matching file
PAC7TA	Tmp dir.: WTA	Output	File containing transactions grouped by type
PAC7MM	User dir.: INPUTMIMA	Output	File containing command MIMA
PAC7MI	User dir.: MIA1	Output	File containing GCP commands
PAC7MO	User dir.: MIA2	Output	File containing GCO commands
PAC7MC	User dir.: MIA3	Output	File containing GGC commands
PAC7MS	User dir.: MIA4	Output	File containing GGS commands

Reorganization of transactions : BVPACS97

Code	Physical name	Type	Label
PAC7TA	Tmp dir.: WTA	Input	File containing transactions grouped by type
PAC7AT	Tmp dir.: WAT	Output	File with transactions grouped by type, sorted again

Sort and formatting of import file: BVPACS91

Code	Physical name	Type	Label
PAC7AT	Tmp dir.: WAT	Input	File containing transactions grouped by type
PAC7TI	Tmp dir.: WTI	Output	File with transactions grouped by type, sorted again

Tri global : BVPACS98

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Output	Development Database data
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7RQ	Tmp dir.: WPAC7QR	Input	Matching Entity/lowest Library
PAC7TI	Tmp dir.: WAT	Input	File with transactions grouped by type, sorted again
PAC7TG	User dir.: MIBJ	Output	Result file with global sort

MIBJ - Execution Script

```
'-----  
'          VISUALAGE PACBASE  
'-----  
'          ARCHIVED JOURNAL MIGRATION  
'-----  
'  
<job id=MIBJ>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "MIBJ"  
</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" , "PACS9B"))  
'-----  
WshEnv("PAC7RN") = BVP_MibrNr  
Call BvpEnv("PACS9B","PAC7NR",Rep_TMP & "\\WNR.tmp")  
Call RunCmdLog ("BVPACS9B")  
Call Err_Cod(Return , 0 , "PACS9B")  
  
Call Msg_Log (Array("1022" , "PACSJR"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
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("PAC7MB") = Fic_Input  
Call BvpEnv("PACSJR","PAC7DD",Rep_USR & "\\MIBJDDSJ.txt")  
Call BvpEnv("PACSJR","PAC7EE",Rep_USR & "\\MIBJEESJ.txt")  
Call BvpEnv("PACSJR","PAC7PJ",Rep_SAVE & "\PJ")  
Call BvpEnv("PACSJR","PAC7RQ",Rep_TMP & "\\WRQJR.tmp")  
Call RunCmdLog ("BVPACSJR")  
Call Err_Cod(Return , 0 , "PACSJR")  
  
Call Msg_Log (Array("1022" , "PACS93"))  
'-----  
WshEnv("PAC7AR") = Rep_BASE & "\AR"  
WshEnv("PAC7AN") = Rep_BASE & "\AN"  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7SL") = BVP_MibrS1  
Call BvpEnv("PACS93","PAC7RQ",Rep_TMP & "\\WRQJR.tmp")  
Call BvpEnv("PACS93","PAC7QR",Rep_TMP & "\\WQR.tmp")  
Call RunCmdLog ("BVPACS93")
```

```

Call Err_Cod(Return , 0 , "PACS93")

Call Msg_Log (Array("1022" , "PACS9A"))
'-----
Call BvpEnv("PACS9A","PAC7NR",Rep_TMP & "\\WNR.tmp")
Call BvpEnv("PACS9A","PAC7QR",Rep_TMP & "\\WQR.tmp")
Call BvpEnv("PACS9A","PAC7RQ",Rep_TMP & "\\WRQ.tmp")
Call RunCmdLog ("BVPACS9A")
Call Err_Cod(Return , 0 , "PACS9A")

Call Msg_Log (Array("1022" , "PACS92"))
'-----
Call BvpEnv("PACS92","PAC7QR",Rep_TMP & "\\WRQ.tmp")
Call BvpEnv("PACS92","PAC7RQ",Rep_TMP & "\\WPACXRQ.tmp")
Call BvpEnv("PACS92","PAC7MZ",Rep_TMP & "\\WMB.tmp")
Call RunCmdLog ("BVPACS92")
Call Err_Cod(Return , 0 , "PACS92")

Call Msg_Log (Array("1022" , "PACX"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\\AE"
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"
Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\\WMB.tmp")
Call BvpEnv("PACX","PAC7BM",Rep_TMP & "\\WBM.tmp")
Call BvpEnv("PACX","PAC7WD",Rep_TMP & "\\WWD.tmp")
Call BvpEnv("PACX","PAC7MM",Rep_TMP & "\\WMM.tmp")
Call BvpEnv("PACX","PAC7MJ",Rep_TMP & "\\WMJ.tmp")
Call BvpEnv("PACX","PAC7TE",Rep_TMP & "\\WTE.tmp")
Call BvpEnv("PACX","PAC7RE",Rep_TMP & "\\WRE.tmp")
Call BvpEnv("PACX","PAC7RQ",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\\WGY.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\\WCP.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\\PACXEZ.txt")

```

```

Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACS94"))
'-----
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACS94","PAC7RQ",Rep_TMP & "\WPACXRQ.tmp")
Call BvpEnv("PACS94","PAC7QR",Rep_TMP & "\WPAC7QR.tmp")
Call RunCmdLog ("BVPACS94")
Call Err_Cod(Return , 0 , "PACS94")

Call Msg_Log (Array("1022" , "PACS96"))
'-----
Call BvpEnv("PACS96","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
WshEnv("PAC7PM") = BVP_MibaFi
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACS96","PAC7TA",Rep_TMP & "\WTA.tmp")
Call BvpEnv("PACS96","PAC7MM",Rep_USR & "\INPUTMIMA.txt")
Call BvpEnv("PACS96","PAC7MI",Rep_USR & "\MIA1.txt")
Call BvpEnv("PACS96","PAC7M0",Rep_USR & "\MIA2.txt")
Call BvpEnv("PACS96","PAC7MC",Rep_USR & "\MIA3.txt")
Call BvpEnv("PACS96","PAC7MS",Rep_USR & "\MIA4.txt")
Call RunCmdLog ("BVPACS96")
Call Err_Cod(Return , 0 , "PACS96")

Call Msg_Log (Array("1022" , "PACS97"))
'-----
Call BvpEnv("PACS97","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACS97","PAC7TA",Rep_TMP & "\WTA.tmp")
Call RunCmdLog ("BVPACS97")
Call Err_Cod(Return , 0 , "PACS97")

Call Msg_Log (Array("1022" , "PACS91"))
'-----
Call BvpEnv("PACS91","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACS91","PAC7TI",Rep_TMP & "\WTI.tmp")
Call RunCmdLog ("BVPACS91")
Call Err_Cod(Return , 0 , "PACS91")

Call Msg_Log (Array("1022" , "PACS98"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
CALL BVPENV("PACS98","PAC7CP",Rep_TMP & "\WCP.TMP")
Call BvpEnv("PACS98","PAC7TI",Rep_TMP & "\WTI.tmp")
Call BvpEnv("PACS98","PAC7RQ",Rep_TMP & "\WPAC7QR.tmp")
Call BvpEnv("PACS98","PAC7TG",Rep_USR & "\MIBJ.ta2")

```

```

Call RunCmdLog ("BVPACS98")
Call Err_Cod(Return , 0 , "PACS98")

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

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

</script>
</job>

```

MIMA - Migration of the Macros

MIMA - Introduction

This procedure allows to simultaneously generate and tag one or more Macrostructures at once.

You can indicate the input of this procedure in two ways:

- Either indicate a '*' line (user code, password, library...) as input, as well as one or more command lines for each Macrostructure to be generated.
- or use the INPUTMIMA file produced by the MIBA, MIBR or MIBJ procedures. This file contains one or more GCM command lines for each Macrostructure to be generated. It may also contain GMO command lines if high priority Macrostructures have been generated (1).

Caution: You must not modify the command lines of the INPUTMIMA file.

(1) These Macrostructures, which are on 6 characters (Screen code), will be generated on 8 characters with an 'SP' suffix (SCREENSP). They will contain only the N-type lines with the relative positions.

MIMA - User Input

A '*' line with the user code, password and Library code is required.

Pos.	Lon.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code
22	4	nnnn	Session number (blank=current ses.)
26	1	'T'	Session status if frozen session

Then you must enter a line for the GCM command, and enter parameters (optional).

Pos.	Lon.	Value	Meaning
2	1	'Z'	Line code
3	2	'90'	Printing order criterion (1)
5	4	'GCM '	Command code (1)
9	6	cccccc	Macrostructure code (1)
15	1	'C'	Selection indicator (1)
16	1	'1'	Option to be printed (1)
28	1		Presence of a continuation line
		' '	No continuation line
		'*'	Continuation line
31	50		Command parameters

(1) Parameters to be entered on the first line only, not on the continuation lines.

This procedure also allows to replace unknown special characters in a Macrostructure code with other characters or to replace unauthorized words (Windows reserved words for example) with other codes. The matching table is the PAC7CS file used as input to the BVPACP87 program (optional file, created by the MIAM procedure and updated by the client if necessary).

MIMA - Description of Steps

Generation/Print commands : BVPACA10

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AY	Database dir.: AY	Input	Development Database extension
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7ME	User Input	Input	User transactions

Code	Physical name	Type	Label
PAC7MV	Tmp dir.: WMV	Output	Update transactions
PAC7MG	Tmp dir.: WMG	Output	Generation/Print commands

GCM specific preparation : BVPACA30

Code	Physical name	Type	Label
PAC7MG	Tmp dir.: WMG	Input	Generation/Print commands
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel dir.: AE	Input	Error messages
PAC7JG	Tmp dir.: WKC	Output	Extractor command
PAC7KU	Tmp dir.: WKU	Output	Report

Macro generation: BVPACMM

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - DBase dir.: AN	Input	Administration Database data
PACGGN	Admin. Database - DBase dir.: AR	Input	Administration Database index
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AY	Database dir.: AY	Input	Development Database extension data
PAC7SC	System - Skel. dir.: SC	Input	Generation skeleton
PAC7CS	User dir.: MibaSpecChar	Input	Special characters matching file
PAC7EP	Tmp dir.: WEP	Output	Work file
PAC7GI	Tmp dir.: WGI	Output	Work file
PAC7GP	Tmp dir.: WGP	Output	Work file
PAC7JG	Tmp dir.: WJG	Output	Work file
PAC7KC	Tmp dir.: WKC	Output	Work file
PAC7KP	Tmp dir.: WKP	Output	Work file

Code	Physical name	Type	Label
PAC7RQ	Tmp dir.: WRQ	Output	Work file
PAC7AT	Tmp dir.: WAT	Output	Work file
PAC7TA	Tmp dir.: WTA	Output	Work file
PAC7TT	User dir.: MIMA	Output	Generated Macros file
PAC7WA	Tmp dir.: WWA	Output	Work file
PAC7WC	Tmp dir.: WWC	Output	Work file
PAC7W1	Tmp dir.: WW1	Output	Work file
PAC7W2	Tmp dir.: WW2	Output	Work file
PAC7W3	Tmp dir.: WW3	Output	Work file
PAC7W4	Tmp dir.: WW4	Output	Work file
PAC7W5	Tmp dir.: WW5	Output	Work file
PAC7IA	User dir.: MIMAIID	Report	General printout of the program stream

MIMA - Execution Script

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          MACRO-STRUCTURES MIGRATION          |
|-----|
|
<job id=MIMA>

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

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

<script language="VBScript">

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

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

Call Msg_Log (Array("1022" , "PACA10"))

```



```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA30"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PACA30","PAC7KU",Rep_TMP & "\WКУ.tmp")
CALL BVPENV("PACA30","PAC7JG",REP_TMP & "\WKC.TMP")
Call BvpEnv("PACA30","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA30")
Call Err_Cod(Return , 0 , "PACA30")

Call Msg_Log (Array("1022" , "PACMM"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7SC") = Rep_SKEL & "\SC" & CodLang
Call BvpEnv("PACMM","PAC7EP",Rep_TMP & "\WEP.tmp")
Call BvpEnv("PACMM","PAC7GI",Rep_TMP & "\WGI.tmp")
Call BvpEnv("PACMM","PAC7GP",Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM","PAC7KC",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM","PAC7KP",Rep_TMP & "\WKP.tmp")
Call BvpEnv("PACMM","PAC7RQ",Rep_TMP & "\WRQ.tmp")
Call BvpEnv("PACMM","PAC7AT",Rep_TMP & "\WAT.tmp")
Call BvpEnv("PACMM","PAC7TA",Rep_TMP & "\WTA.tmp")
Call BvpEnv("PACMM","PAC7TT",Rep_USR & "\MIMA.cb1gen")
Call BvpEnv("PACMM","PAC7WA",Rep_TMP & "\WWA.tmp")
Call BvpEnv("PACMM","PAC7WC",Rep_TMP & "\WWC.tmp")
Call BvpEnv("PACMM","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM","PAC7W3",Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACMM","PAC7W4",Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACMM","PAC7W5",Rep_TMP & "\WW5.tmp")
WshEnv("PAC7CS") = BVP_MibaCs
Call BvpEnv("PACMM","PAC7IA",Rep_USR & "\MIMAI.A.txt")
Call RunCmdLog ("BVPACMM")

```

```

Call Err_Cod(Return , 0 , "PACMM")

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

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

</script>
</job>

```

MITH - Migration of the Thesaurus

MITH - Introduction

Principle

This procedure is a utility program that is used to migrate the thesaurus in batch mode. After the extraction of entities in inter-library mode in the repository, an XML-formatted file, containing the thesaurus, will be exported to the SDP.

MITH - User Input

The user input of the MITH procedure is identical to that of the PACX procedure, with the following specific values:

One '*' line as follows:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	***	Extraction inter-library code
22	4	nnnn	Session number (Blank=current session)
26	1	'T'	Session status if frozen session
29	4	EXLI	Extractor code
34	1	'1'	Formatting for UPDP (PAF)
55	1	'1'	Convergence plan flag

The EXLI extraction is executed in inter-library, because the thesaurus entities are created in inter-library within the repository.

The GY file resulting from PACX extraction is then filtered, the thesaurus records are selected and a new XML-formatted file is created.

MITH - Description of Steps

Extraction: PACX

This step extracts transactions according to user input.

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AN	Database dir. : AN	Input	Development Database index
PAC7AR	Database dir. : AR	Input	Development Database data
PAC7AY	Database dir. : AY	Input	Development Database extension data
PACGGN	Admin Database - DBase dir. : AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7MB	User input	Input	User input
PAC7MA	NUL	Input	Work file
PAC7ES	NUL	Input	Work file
PAC7BM	Tmp dir.: WBM	Input/Output	User input
PAC7MM	Tmp dir.: WMM	Input/Output	Work file
PAC7MJ	Tmp dir.: WMJ	Input/Output	Work file
PAC7TE	Tmp dir.: WTE	Input/Output	Work file
PAC7RE	Tmp dir.: WRE	Input/Output	Work file
PAC7RM	Tmp dir.: WRM	Input/Output	Work file
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted transactions
SYSEXT	Tmp dir.: WSY	Input/Output	Work file
PAC7MV	Tmp dir.: WMV	Output	Work file
PAC7MR	Tmp dir.: WMR	Output	Work file
PAC7MX	Tmp dir.: WMX	Output	Entities not extracted (PACX)
PAC7GY	Tmp dir.: : WGY	Output	Extracted transactions for UPDP

Code	Physical name	Type	Label
PAC7RQ	NUL	Output	Matching Entity/initial Library
PAC7TD	Tmp dir.: WTD	Output	Work file
PAC7UE	Tmp dir.: WUE	Output	Work file
PAC7CP	NULCP	Output	
PAC7IA	User dir.: PACXIA	Report	General printout of the program stream
PAC7DD	User dir.: PACXDD	Report	Errors on input transactions
PAC7ED	Tmp dir.: WED	Report	Extraction report
PAC7EE	User dir.: PACXEE	Report	Extraction report
PAC7EG	Tmp dir.: WEG	Report	Extraction report
PAC7EM	Tmp dir.: WEM	Report	Extraction report
PAC7EP	User dir.: PACXEP	Report	Extraction report
PAC7EQ	User dir.: PACXEQ	Report	Extraction report
PAC7EU	Tmp dir.: WEU	Report	Extraction report
PAC7EZ	User dir.: PACXEZ	Report	Extraction report

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD)

Selection of thesaurus transactions and formatting of an XML file :
BVPACTH1

Code	Physical name	Type	Label
PAC7GY	Tmp dir.: WGY	Input	File resulting from PACX extraction
PAC7RQ	User dir.: MITHFILE	Output	Thesaurus file in XML format

MITH - Execution Script

```

|-----|
| VISUALAGE PACBASE |
|-----|
|          THESAURUS  MIGRATION          |
|-----|

```

```

'
<job id=MITH>

<script language="VBScript">
Dim MyProc
MyProc = "MITH"
</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" , "PACX"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7MB") = Fic_Input
Call BvpEnv("PACX", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACX", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("PACX", "PAC7MM", Rep_TMP & "\WMM.tmp")
Call BvpEnv("PACX", "PAC7MJ", Rep_TMP & "\WMJ.tmp")
Call BvpEnv("PACX", "PAC7TE", Rep_TMP & "\WTE.tmp")
Call BvpEnv("PACX", "PAC7RE", Rep_TMP & "\WRE.tmp")
Call BvpEnv("PACX", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7RM", Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX", "PAC7MA", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7ES", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX", "PAC7CP", Rep_TMP & "\NULCP.tmp")
Call BvpEnv("PACX", "PAC7UE", Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX", "PAC7TD", Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX", "PAC7IA", Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX", "PAC7DD", Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX", "PAC7ED", Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX", "PAC7EE", Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX", "PAC7EG", Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX", "PAC7EM", Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX", "PAC7EP", Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX", "PAC7EQ", Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX", "PAC7EU", Rep_TMP & "\WEU.tmp")
Call BvpEnv("PACX", "PAC7EZ", Rep_USR & "\PACXEZ.txt")
Call BvpEnv("PACX", "SYSEXT", Rep_TMP & "\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))

```

```

End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PACTH1"))
'-----
Call BvpEnv("PACTH1","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACTH1","PAC7RQ",Rep_USR & "\MITHFILE.txt")
Call RunCmdLog ("BVPACTH1")
Call Err_Cod(Return , 0 , "PACTH1")

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

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

</script>
</job>

```

MIUS - Migration of the Users

MIUS - Introduction

Principle

Transformation of users into create user.

This procedure is a utility that transforms Users into Create users. After the extraction of the users declared in the administration, a file containing the characteristics of each user is created. This file contains a list of user IDs, email addresses, names, license IDs and administration level, separated by commas. This file is used as input to the RTC reptools tool.

MIUS - User Input

The user input is the one of the PACX procedure with the following specific values:

One '*' line as follows:

Pos.	Len.	Value	Meaning
2	1	*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	***	Extraction inter-library code

Pos.	Len.	Value	Meaning
22	4		Session number (Blank=current)
26	1		Session status
29	4	EXTR	Extractor code
34	1	'1'	Formatting for UPDP (PAF)
55	1	'1'	Convergence plan flag

A command line:

Pos.	Len.	Value	Meaning
2	1	'W'	Line code
3	1	'1'	Line number
4	2	'EX'	
6	1		Library selection code
		'C'	Library and its upper-libraries
7	33	1USE	Users Meta-entity
40	4		Extraction type (1):
		'ALL '	Entity and used entities

The EXTR extraction is executed in inter-libraries, in the Administration Database.

The GY file resulting from PACX is then filtered, the users records are selected and a new file, used as input to the RTC repotools tool, is created.

Note:

The commas delimit the fields. When they are inside a text field, they are replaced with semi-colons in order to be compatible with RTC.

MIUS - Description of Steps

Extraction : PACX

This step extracts transactions according to user input.

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AN	Admin Database - Base dir. : AN	Input	Administration Database index

Code	Physical name	Type	Label
PAC7AR	Admin Database - Base dir. : AR	Input	Administration Database data
PAC7AY	Admin Database - Base dir. : AY	Input	Administration Database extension data
PACGGN	Admin Database - Base dir. : AN	Input	Administration Database index
PACGGR	Admin Database - Base dir. : AR	Input	Administration Database data
PACGGU	Admin Database - Base dir. : GU	Input	Administration Database users
PAC7PJ	Admin Database - Save dir. : PJ	Input	Archived transactions
PAC7MB	User input	Input	User input
PAC7BM	Tmp dir. : WBM	Input/Output	User input
PAC7MM	Tmp dir. : WMM	Input/Output	
SYSEXT	Tmp dir. : SY	Input/Output	Work file
PAC7MJ	Tmp dir. : WMJ	Input/Output	
PAC7TE	Tmp dir. : WTE	Input/Output	
PAC7RE	Tmp dir. : WRE	Input/Output	
PAC7RM	Tmp dir. : WRM	Input/Output	
PAC7RQ	Tmp dir. : WRQ	Input/Output	Work file
PAC7WD	Tmp dir. : WWD	Input/Output	Extracted transactions
PAC7MV	Tmp dir. : WMV	Input/Output	
PAC7MR	Tmp dir. : WMR	Input/Output	
PAC7TD	Tmp dir. : WTD	Input/Output	
PAC7MX	Tmp dir. : WMX	Input/Output	
PAC7GY	Tmp dir. : WGY	Output	Extracted transactions for UPGP
PAC7UE	Tmp dir. : WUE	Input/Output	
PAC7IA	User dir. : PACXIA	Report	General printout of the program stream
PAC7DD	User dir. : PACXDD	Report	Errors on input transactions
PAC7ED	Tmp dir. : WED	Report	Extraction report
PAC7EE	User dir. : PACXEE	Report	Extraction report
PAC7EG	Tmp dir. : WEG	Report	Extraction report
PAC7EM	Tmp dir. : WEM	Report	Extraction report

Code	Physical name	Type	Label
PAC7EP	User dir. : PACXEP	Report	Extraction report
PAC7EQ	User dir. : PACXEQ	Report	Extraction report
PAC7ES	NUL		
PAC7EU	Tmp dir. : WEU	Report	Extraction report
PAC7EZ	User dir. : PACXEZ	Report	Extraction report
PAC7MA	NUL		
PAC7CP	NULCP		

Return codes:

- 0 : No error
- 4 : Error in user input (detailed in PAC7EE) or in the extraction
- 8 : Error in '*' line (detailed in PAC7DD)

Transformation of users into create user : BVPLTYAU

Code	Physical name	Type	Label
PAC7GY	Tmp dir. : WGY	Input	File resulting from PACX extraction
PAC7US	User dir. : MIUSFILE	Output	Users file: input of repotools tool

MIUS - Execution Script

```

| -----
|          VISUALAGE PACBASE
|
| -----
|                   USERS  MIGRATION
|
| -----
|
<job id=MIUS>

<script language="VBScript">
Dim MyProc
MyProc = "MIUS"
</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" , "PACX"))

```

```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7PJ") = Rep_ASARE & "\PJ"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PACX","PAC7BM",Rep_TMP & "\WBM.tmp")
Call BvpEnv("PACX","PAC7WD",Rep_TMP & "\WWD.tmp")
Call BvpEnv("PACX","PAC7MM",Rep_TMP & "\WMM.tmp")
Call BvpEnv("PACX","PAC7MJ",Rep_TMP & "\WMJ.tmp")
Call BvpEnv("PACX","PAC7TE",Rep_TMP & "\WTE.tmp")
Call BvpEnv("PACX","PAC7RE",Rep_TMP & "\WRE.tmp")
Call BvpEnv("PACX","PAC7RQ",Rep_TMP & "\WRQ.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\NULCP.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\WEU.tmp")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\PACXEZ.txt")
Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If

If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PLTYAU"))
'-----
Call BvpEnv("PLTYAU","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PLTYAU","PAC7US",Rep_USR & "\MIUSFILE.txt")
Call RunCmdLog ("BVPLTYAU")
Call Err_Cod(Return , 0 , "PLTYAU")

```

```

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

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

</script>
</job>

```

MILA - Migration of Administration Error Messages

MILA - Introduction

Principle

Extraction of the customized error messages of the administration.

This procedure is a utility that formats an error messages file. This file contains all the standard error messages. An extraction of customized error messages in the Administration Database allows to override the corresponding standard error messages. The output file contains all the error messages for all languages, each set is separated by a comment line indicating the language.

MILA - User Input

The user input is the one of the PACX procedure with the following specific values:

One '*' line as follows:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	***	Extraction inter-library code
22	4		Session number (Blank=current)
26	1		Session status
29	4	EXTR	Extractor code
34	1	'1'	Formatting for UPDP (PAF)
55	1	'1'	Convergence plan flag

A command line:

Pos.	Len.	Value	Meaning
2	1	'W'	Line code
3	1	'1'	Line number
4	2	'EX'	
6	1		Library selection code
		'C'	Library and its upper-libraries
7	33	YATTABLES	User entity of administration tables

The EXTR extraction is executed in inter-library mode, in the Administration Database.

The GY file resulting from PACX is then filtered, the error messages records are selected (-D02 description) and formatted again in a PAC7LI file.

MILA - Description of Steps

Extraction : PACX

This step extracts transactions according to user input.

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AN	Admin Database - Base dir. : AN	Input	Administration Database index
PAC7AR	Admin Database - Base dir. : AR	Input	Administration Database data
PAC7AY	Admin Database - Base dir. : AY	Input	Administration Database extension data
PACGGN	Admin Database - Base dir. : AN	Input	Administration Database index
PACGGR	Admin Database - Base dir. : AR	Input	Administration Database data
PACGGU	Admin Database - Base dir. : GU	Input	Administration Database users
PAC7PJ	Admin Database - Save dir. : PJ	Input	Archived transactions
PAC7MB	User input	Input	User input
PAC7BM	Tmp dir. : WBM	Input/Output	User input
PAC7MM	Tmp dir. : WMM	Input/Output	
SYSEXT	Tmp dir. : SY	Input/Output	Work file

Code	Physical name	Type	Label
PAC7MJ	Tmp dir. : WMJ	Input/Output	
PAC7TE	Tmp dir. : WTE	Input/Output	
PAC7RE	Tmp dir. : WRE	Input/Output	
PAC7RM	Tmp dir. : WRM	Input/Output	
PAC7RQ	Tmp dir. : WRQ	Input/Output	Work file
PAC7WD	Tmp dir. : WWD	Input/Output	Extracted transactions
PAC7MV	Tmp dir. : WMV	Input/Output	
PAC7MR	Tmp dir. : WMR	Input/Output	
PAC7TD	Tmp dir. : WTD	Input/Output	
PAC7MX	Tmp dir. : WMX	Input/Output	
PAC7GY	Tmp dir. : WGY	Output	Extracted transactions for UPGP
PAC7UE	Tmp dir. : WUE	Input/Output	
PAC7IA	User dir. : PACXIA	Report	General printout of the program stream
PAC7DD	User dir. : PACXDD	Report	Errors on input transactions
PAC7ED	Tmp dir. : WED	Report	Extraction report
PAC7EE	User dir. : PACXEE	Report	Extraction report
PAC7EG	Tmp dir. : WEG	Report	Extraction report
PAC7EM	Tmp dir. : WEM	Report	Extraction report
PAC7EP	User dir. : PACXEP	Report	Extraction report
PAC7EQ	User dir. : PACXEQ	Report	Extraction report
PAC7ES	NUL		
PAC7EU	Tmp dir. : WEU	Report	Extraction report
PAC7EZ	User dir. : PACXEZ	Report	Extraction report
PAC7MA	NUL		
PAC7CP	NULCP		

Return codes:

- 0 : No error
- 4 : Error in user input (detailed in PAC7EE) or in the extraction
- 8 : Error in '*' line (detailed in PAC7DD)

Formatting of error messages: BVPLTYAT

Code	Physical name	Type	Label
PAC7GY	Tmp dir. : WGY	Input	File resulting from PACX extraction
PAC7LI	User dir. : MILAFILE	Output	Users file: input of repotools tool

MILA - Execution Script

```

|-----|
|          VISUALAGE PACBASE          |
|-----|
|          ERROR LABEL MIGRATION      |
|-----|
|
|<job id=MILA>
|
|<script language="VBScript">
|Dim MyProc
|MyProc = "MILA"
|</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" , "PACX"))
|-----|
|WshEnv("PAC7AE") = Rep_SKEL & "\AE"
|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("PAC7PJ") = Rep_ASARE & "\PJ"
|WshEnv("PAC7MB") = Fic_Input
|Call BvpEnv("PACX", "PAC7BM", Rep_TMP & "\WBM.tmp")
|Call BvpEnv("PACX", "PAC7WD", Rep_TMP & "\WWD.tmp")
|Call BvpEnv("PACX", "PAC7MM", Rep_TMP & "\WMM.tmp")
|Call BvpEnv("PACX", "PAC7MJ", Rep_TMP & "\WMJ.tmp")
|Call BvpEnv("PACX", "PAC7TE", Rep_TMP & "\WTE.tmp")
|Call BvpEnv("PACX", "PAC7RE", Rep_TMP & "\WRE.tmp")
|Call BvpEnv("PACX", "PAC7RQ", Rep_TMP & "\WRQ.tmp")
|Call BvpEnv("PACX", "PAC7RM", Rep_TMP & "\WRM.tmp")
|Call BvpEnv("PACX", "PAC7MA", Rep_TMP & "\NUL.tmp")
|Call BvpEnv("PACX", "PAC7CP", Rep_TMP & "\NULCP.tmp")
|Call BvpEnv("PACX", "PAC7ES", Rep_TMP & "\NUL.tmp")
|Call BvpEnv("PACX", "PAC7UE", Rep_TMP & "\WUE.tmp")
|Call BvpEnv("PACX", "PAC7GY", Rep_TMP & "\WGY.tmp")
|Call BvpEnv("PACX", "PAC7TD", Rep_TMP & "\WTD.tmp")

```

```

Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\\PACXEZ.txt")
Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PLTYAT"))
'-----
Call BvpEnv("PLTYAT","PAC7GY",Rep_TMP & "\\WGY.tmp")
Call BvpEnv("PLTYAT","PAC7LI",Rep_USR & "\\MILAFIle.txt")
Call RunCmdLog ("BVPLTYAT")
Call Err_Cod(Return , 0 , "PLTYAT")

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

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

</script>
</job>

```

MERT - Transformation of Model entities

MERT - Introduction

Principle

The Model entities (M entities with an Object-, Relation- or Functional Integrity Constraint-type) are not retrieved into RPP. These entities are used in the Pacdesign module only. They may be used to store other information that is not managed with Pacbase.

This procedure extracts the Model entities and transforms them into User entities that can be exported to the SDP.

This procedure produces a file that contains the update transactions to be used as input to the UPDP procedure.

This procedure is not required.

MERT - User input

A '*' with the following information:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	***	Inter-Library extraction
22	4	nnnn	Session number (blank=current)
26	1	'T'	Session status if frozen session

This procedure processes the following lines only:

- Definition of the Method entities (Object, Relationship, and FIC).
- Comment lines: -GC.
- Properties call lines: -CE.
- Comment lines: -GC under -CE.
- Merise objects call lines: -CM.
- Comment lines: -GC under -CM.

The following lines are not processed:

- Input Aid call lines in -GC.
- Merise entities cross-reference lines.

Transformation of the Model entities into User Entities

- Merise objects are transformed into User Entities with a \$9O call type.
- Merise relationships are transformed into User Entities with a \$9R call type.
- Merise FICs are transformed into User Entities with a \$9C call type.

MERT - Description of steps

Definition of the new Meta Entities

- Create the new Meta Entities by running the VINS procedure with the BVPMETHOD file.

- Run the MERT procedure.
- Run the UPDP procedure to update the Pacbase database with the new User Entities created by the MERT procedure.
- Apply the new format to the Pacdesign User Entities associated with the model used by running the VINS procedure with the BVPMERF2 file, if the model used is MERISE.
- Reorganize the database with the REOR procedure to take the index file into account.
CAUTION: After this update, the modified entities cannot be used anymore by Pacdesign.

You can obtain the BVPMETHOD and BVPMERF2 file from the VisualAge Pacbase support.

Extraction of entities: PAFMER

Code	Physical name	Type	Label
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin database - DBase dir.: GU	Input	Administration Database users
PAC7AR	DBase dir.: AR	Input	Development Database data
PAC7AN	DBase dir.: AN	Input	Development Database index
PAC7MB	User input	Input	User input
SYSPAF	Tmp dir.: SYSPAF	Input/Output	PAF indexed file
PAC7UR	Tmp dir.: WUR	Input/Output	Work file
PAC7EJ	User dir.: MERT1EJ	Report	Check lists
PAC7EI	User dir.: MERT1ET	Report	PAF access report
PAC7DD	User dir.: MERT1DD	Report	Error list

Writing of entities: BVPTUMER

Code	Physical name	Type	Label
PAC7UR	Tmp. dir.: WUR	Input	Work file
PAC7GY	User dir.: MERTGY	Output	Merise entities file
PAC7EJ	User dir.: MERT2EJ	Report	Integrity check lists

Code	Physical name	Type	Label
PAC7ET	User dir.: MERT2ET	Report	PAF access file
PAC7DD	User dir.: MERT2DD	Report	Error list

MERT - Execution JCL

```

| -----
|          VISUALAGE PACBASE
| -----
|          - MERISE ENTITY TRANSFORMATION -
| -----
|
<job id=MERT>

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

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

<script language="VBScript">

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

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
  CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

Call Msg_Log (Array("1022" , "PAFMER"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7MB") = Fic_Input
Call BvpEnv("PAFMER", "PAC7UR", Rep_TMP & "\WUR.tmp")
Call BvpEnv("PAFMER", "SYSPAF", Rep_TMP & "\SYSPAF.tmp")
Call BvpEnv("PAFMER", "PAC7EJ", Rep_USR & "\MERT1EJ.txt")
Call BvpEnv("PAFMER", "PAC7ET", Rep_USR & "\MERT1ET.txt")
Call BvpEnv("PAFMER", "PAC7DD", Rep_USR & "\MERT1DD.txt")
Call RunCmdLog ("BVPAFMER")
Call Err_Cod(Return , 0 , "PAFMER")

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

```

```

'-----
Call BvpEnv("PTUMER","PAC7UR",Rep_TMP & "\WUR.tmp")
Call BvpEnv("PTUMER","PAC7GY",Rep_USR & "\MERTGY.txt")
Call BvpEnv("PTUMER","PAC7EJ",Rep_USR & "\MERT2EJ.txt")
Call BvpEnv("PTUMER","PAC7ET",Rep_USR & "\MERT2ET.txt")
Call BvpEnv("PTUMER","PAC7DD",Rep_USR & "\MERT2DD.txt")
Call RunCmdLog ("BVPTUMER")
Call Err_Cod(Return , 0 , "PTUMER")

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

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

</script>
</job>

```

MIAx - Optional control cards

You must define commands in the Generation Manager to split the generated Programs, Screens, Servers, or "COPY" clauses into separate files and into directories that you specify.

Definition of the command lines IN FRONT.

These lines contain the character strings specific to these lines, but also information on the file to be produced: its name, extension, and the directory in which it is expected.

The first command line IN FRONT must contain:

Value	Meaning
++++	Delimiter, always in columns 1 to 7
PGMCOD	Name of the file to be produced
ext	Extension, on 3 characters maximum

The values are separated by a blank.

Example:

```
*++++* MYPROGR cb1
```

The second command line IN FRONT must contain:

Value	Meaning
&&&&&	Delimiter, always in columns 1 to 7
path	Path for the writing of files

The values are separated by a blank.

Example:

&&&&& /vapac/cobol

MIA1 - Program Generation

MIA1 - Introduction

This procedure enables you to generate the programs in two files: a file that contains the generated COBOL code and another file that contains the control COBOL code.

It generates the micropatterns which do not come from Macros.

The generation is carried out without line numbers.

The right part (columns 73 to 80) is generated as a blank, except for the function tag lines where lvnn is generated (with nn = level of the function/subfunction).

MIA1 - User Input

A '*' line with the user code, password and Library code is required.

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code

You must then specify a GCP command line as described in the input of the GPRT procedure for each program to be generated.

MIA1 - Description of Steps

Generation/Print commands : BVPACA10

Code	Physical name	Type	Label
PAC7AR	DBase dir.: AR	Input	Development Database data

Code	Physical name	Type	Label
PAC7AN	DBase dir.: AN	Input	Development Database index
PAC7AY	DBase dir.: AY	Input	Development Database extension
PAC7AJ	DBase dir.: AJ	Input	Development Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin Database - DBase dir. AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir.: GU	Input	Administration Database users
PAC7ME	User Input	Input	User transactions
PAC7MV	Tmp dir.: WMV	Output	Update transactions
PAC7MG	Tmp dir.: WMG	Output	Generation/Print commands

Dispatch of command lines : BVPACA20

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PACGGK	Admin. Database - DBase dir.: GK	Input/ Output	Generation rights
PAC7JG	Tmp dir.: WJG	Output	Extractor command
PAC7KF	Tmp dir.: WKF	Output	Work file
PAC7MG	Tmp dir.: WMG	Input	Generation/Print commands
PAC7W1	NUL	Input/ Output	Work file
PAC7KU	Tmp dir.: WKU	Output	Output report

Program generation : BVPACMM1

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AY	Database dir.: AY	Input	Development Database extension data
PAC7SC	System - Skel. dir.: SC	Input	Generation skeleton
PAC7EP	Tmp dir.: WEP	Output	Work file
PAC7GI	Tmp dir.: NUL	Output	Work file
PAC7GP	Tmp dir.: WGP	Output	Work file
PAC7JG	Tmp dir.: WJG	Output	Work file
PAC7KC	Tmp dir.: WKC	Output	Work file
PAC7KP	Tmp dir.: WKP	Output	Work file
PAC7G1	Tmp dir.: WG1	Output	Work file
PAC7G2	User dir.: COBOL_MIA1	Output	C9 COBOL file
PAC7G3	User dir.: CONTROL_COBOL_MIA1	Output	Control COBOL file
PAC7WB	Tmp dir.: WWB	Output	Work file
PAC7WC	Tmp dir.: WWC	Output	Work file
PAC7WM	Tmp dir.: WWM	Output	Work file
PAC7WP	Tmp dir.: WWP	Output	Work file
PAC7W1	Tmp dir.: WW1	Output	Work file
PAC7W2	Tmp dir.: WW2	Output	Work file
PAC7W3	Tmp dir.: WW3	Output	Work file
PAC7W4	Tmp dir.: WW4	Output	Work file
PAC7W5	Tmp dir.: WW5	Output	Work file
PAC7W6	Tmp dir.: WW6	Output	Work file
PAC7CA	Tmp dir.: WCA	Output	Work file
PAC7CM	Tmp dir.: WCM	Output	Work file

Code	Physical name	Type	Label
PAC7IA	User dir.: MIA1IA	Report	General printout of the program stream

MIA1 - Execution Script

```

| -----
|          VISUALAGE PACBASE
| -----
|          PROGRAM GENERATION
| -----
|
<job id=MIA1>

<script language="VBScript">
Dim MyProc
MyProc = "MIA1"
</script>

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

<script language="VBScript">

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

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

Call Msg_Log (Array("1022" , "PACA10"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"

```

```

WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20", "PAC7JG", Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20", "PAC7KF", Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20", "PAC7KU", Rep_TMP & "\WKU.tmp")
Call BvpEnv("PACA20", "PAC7MG", Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20", "PAC7W1", Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM1"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7SC") = Rep_SKEL & "\SC" & CodLang
Call BvpEnv("PACMM1", "PAC7CA", Rep_TMP & "\WCA.tmp")
Call BvpEnv("PACMM1", "PAC7CM", Rep_TMP & "\WCM.tmp")
Call BvpEnv("PACMM1", "PAC7EP", Rep_TMP & "\WEP.tmp")
Call BvpEnv("PACMM1", "PAC7GI", Rep_TMP & "\NUL.tmp")
Call BvpEnv("PACMM1", "PAC7GP", Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM1", "PAC7JG", Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM1", "PAC7KC", Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM1", "PAC7KP", Rep_TMP & "\WKP.tmp")
Call BvpEnv("PACMM1", "PAC7G1", Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM1", "PAC7G2", Rep_USR & "\COBOL_MIA1.txt")
Call BvpEnv("PACMM1", "PAC7G3", Rep_USR & _
"\CONTROL_COBOL_MIA1.txt")
Call BvpEnv("PACMM1", "PAC7WB", Rep_TMP & "\WVB.tmp")
Call BvpEnv("PACMM1", "PAC7WC", Rep_TMP & "\WVC.tmp")
Call BvpEnv("PACMM1", "PAC7WM", Rep_TMP & "\WVM.tmp")
Call BvpEnv("PACMM1", "PAC7WP", Rep_TMP & "\WVP.tmp")
Call BvpEnv("PACMM1", "PAC7W1", Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM1", "PAC7W2", Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM1", "PAC7W3", Rep_TMP & "\WW3.tmp")
Call BvpEnv("PACMM1", "PAC7W4", Rep_TMP & "\WW4.tmp")
Call BvpEnv("PACMM1", "PAC7W5", Rep_TMP & "\WW5.tmp")
Call BvpEnv("PACMM1", "PAC7W6", Rep_TMP & "\WW6.tmp")
Call BvpEnv("PACMM1", "PAC7IA", Rep_USR & "\MIA1IA.txt")
Call RunCmdLog ("BVPACMM1")
Call Err_Cod(Return , 0 , "PACMM1")

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

Call Msg_Log (Array("1023"))
'-----

```



```
Wscript.Quit (Return)
```

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

MIA2 - Screen Generation

MIA2 - Introduction

This procedure enables you to generate the Screens in two files: a file that contains the generated COBOL code and another file that contains the control COBOL code.

It generates the micropatterns which do not come from Macros.

The generation is carried out without line numbers.

The right part (columns 73 to 80) is generated as a blank, except for the function tag lines where lvnn is generated (with nn = level of the function/subfunction).

MIA2 - User Input

A '*' line with the user code, password and Library code is required.

Pos.	Len.	Value	Meaning
2	1	*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code

You must then specify a GCO command line as described in the input of the GPRT procedure for each screen to be generated.

MIA2 - Description of Steps

Generation/Print commands : BVPACA10

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

Code	Physical name	Type	Label
PACGGN	Admin Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir.: GU	Input	Administration Database users
PAC7ME	User Input	Input	User transactions
PAC7MV	Tmp dir.: WMV	Output	Update transactions
PAC7MG	Tmp dir.: WMG	Output	Generation/Print commands

Dispatch of command lines : BVPACA20

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PACGGK	Admin. Database - DBase dir.: GK	Input/Output	Generation rights
PAC7JG	Tmp dir.: WJG	Output	Extractor command
PAC7KF	Tmp dir.: WKF	Output	Work file
PAC7MG	Tmp dir.: WMG	Input	Generation/Print commands
PAC7W1	NUL	Input/Output	Work file
PAC7KU	Tmp dir.: WKU	Output	Output report

Screen generation : BVPACMM2

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data

Code	Physical name	Type	Label
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AY	Database dir.: AY	Input	Development Database extension data
PAC7SG	System - Skel. dir.: SG	Input	Generation skeleton
PAC7EE	Tmp dir.: WEE	Input	Work file
PAC7GE	Tmp dir.: WGE	Output	Work file
PAC7JG	Tmp dir.: WJG	Output	Work file
PAC7KC	Tmp dir.: WKC	Output	Work file
PAC7KE	Tmp dir.: WKE	Output	Work file
PAC7G1	Tmp dir.: WG1	Output	Work file
PAC7G2	User dir.: COBOL_MIA2	Output	C9 COBOL file
PAC7G3	User dir.: CONTROL_COBOL_MIA2	Output	Control COBOL file
PAC7W1	Tmp dir.: WW1	Output	Work file
PAC7W2	Tmp dir.: WW2	Output	Work file
PAC7EZ	Tmp dir.: WEZ	Output	Work file
PACM2W	Tmp dir.: W2W	Output	Work file
PACM2P	Tmp dir.: W2P	Output	Work file
PAC2GW	Tmp dir.: GW	Output	Work file
PAC2GP	Tmp dir.: WGP	Output	Work file
PAC780	Tmp dir.: W80	Output	Work file
PAC7IA	User dir.: MIA2IA	Report	General printout of the program stream

MIA2 - Execution Script

```

-----
| VISUALAGE PACBASE
|
|-----
| SCREEN GENERATION
|
|-----
|

```

```

<job id=MIA2>

<script language="VBScript">
Dim MyProc
MyProc = "MIA2"
</script>

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

<script language="VBScript">

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

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

Call Msg_Log (Array("1022" , "PACA10"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20","PAC7JG",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20","PAC7KU",Rep_TMP & "\WКУ.tmp")
Call BvpEnv("PACA20","PAC7MG",Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20","PAC7W1",Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM2"))
'-----

```

```

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7SG") = Rep_SKEL & "\SG" & CodLang
Call BvpEnv("PACMM2","PAC7EE",Rep_TMP & "\WE.tmp")
Call BvpEnv("PACMM2","PAC7EZ",Rep_TMP & "\WEZ.tmp")
Call BvpEnv("PACMM2","PAC7GE",Rep_TMP & "\WGE.tmp")
Call BvpEnv("PACMM2","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM2","PAC7KC",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM2","PAC7KE",Rep_TMP & "\WKE.tmp")
Call BvpEnv("PACMM2","PAC7G1",Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM2","PAC7G2",Rep_USR & "\COBOL_MIA2.txt")
Call BvpEnv("PACMM2","PAC7G3",Rep_USR & _
"\CONTROL_COBOL_MIA2.txt")
Call BvpEnv("PACMM2","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM2","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM2","PACM2P",Rep_TMP & "\W2P.tmp")
Call BvpEnv("PACMM2","PACM2W",Rep_TMP & "\W2W.tmp")
Call BvpEnv("PACMM2","PAC2GP",Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM2","PAC2GW",Rep_TMP & "\WGW.tmp")
Call BvpEnv("PACMM2","PAC780",Rep_TMP & "\W80.tmp")
Call BvpEnv("PACMM2","PAC7IA",Rep_USR & "\MIA2IA.txt")
Call RunCmdLog ("BVPACMM2")
Call Err_Cod(Return , 0 , "PACMM2")

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

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

</script>
</job>

```

MIA3 - Client Generation

MIA3 - Introduction

This procedure enables you to generate the Clients in two files: a file that contains the generated COBOL code and another file that contains the control COBOL code.

It generates the micropatterns which do not come from Macros.

The generation is carried out without line numbers.

The right part (columns 73 to 80) is generated as a blank, except for the function tag lines where lvnn is generated (with nn = level of the function/subfunction).

MIA3 - User Input

A '*' line with the user code, password and Library code is required.

Pos.	Lon.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code

You must then specify a GGC command line as described in the input of the GPRT procedure for each client to be generated.

MIA3 - Description of Steps

Generation/Print commands : BVPACA10

Code	Physical name	Type	Label
PAC7AR	DBase dir.: AR	Input	Development Database data
PAC7AN	DBase dir.: AN	Input	Development Database index
PAC7AY	DBase dir.: AY	Input	Development Database extension
PAC7AJ	DBase dir.: AJ	Input	Development Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir.: GU	Input	Administration Database users
PAC7ME	User Input	Input	User transactions
PAC7MV	Tmp dir.: WMV	Output	Update transactions
PAC7MG	Tmp dir.: WMG	Output	Generation/Print commands

Dispatch of command lines : BVPACA20

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index

Code	Physical name	Type	Label
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PACGGK	Admin. Database - DBase dir.: GK	Input/Output	Generation rights
PAC7JG	Tmp dir.: WJG	Output	Extractor command
PAC7KF	Tmp dir.: WKF	Output	Work file
PAC7MG	Tmp dir.: WMG	Input	Generation/Print commands
PAC7W1	NUL	Input/Output	Work file
PAC7KU	Tmp dir.: WKU	Output	Output report

Client generation : BVPACMM3

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AY	Database dir.: AY	Input	Development Database extension data
PAC7SS	System - Skel. dir.: SS	Input	Generation skeleton
PAC7EG	Tmp dir.: WEG	Input	Work file
PAC7GG	Tmp dir.: WGG	Output	Work file
PAC7JG	Tmp dir.: WJG	Output	Work file
PAC7KC	Tmp dir.: WKC	Output	Work file
PAC7KG	Tmp dir.: WKG	Output	Work file

Code	Physical name	Type	Label
PAC7G1	Tmp dir.: WG1	Output	Work file
PAC7G2	User dir.: COBOL_MIA3	Output	C9 COBOL file
PAC7G3	User dir.: CONTROL_COBOL_MIA3	Output	Control COBOL file
PAC7W1	Tmp dir.: WW1	Output	Work file
PAC7W2	Tmp dir.: WW2	Output	Work file
PAC7EZ	Tmp dir.: WEZ	Output	Work file
PACM2W	Tmp dir.: W2W	Output	Work file
PACM2P	Tmp dir.: W2P	Output	Work file
PAC2GW	Tmp dir.: WGW	Output	Work file
PAC2GP	Tmp dir.: WGP	Output	Work file
PAC780	Tmp dir.: W80	Output	Work file
PAC7IA	User dir.: MIA3IA	Report	General printout of the program stream

MIA3 - Execution Script

```

| -----
|          VISUALAGE PACBASE
|
| -----
|          CLIENT GENERATION
|
| -----
|
<job id=MIA3>

<script language="VBScript">
Dim MyProc
MyProc = "MIA3"
</script>

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

<script language="VBScript">

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

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

```



```

Call Msg_Log (Array("1022" , "PACA10"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20","PAC7JG",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20","PAC7KU",Rep_TMP & "\WКУ.tmp")
Call BvpEnv("PACA20","PAC7KF",Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20","PAC7MG",Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20","PAC7W1",Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM3"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7SS") = Rep_SKEL & "\SS" & CodLang
Call BvpEnv("PACMM3","PAC7EG",Rep_TMP & "\WEG.tmp")
Call BvpEnv("PACMM3","PAC7EZ",Rep_TMP & "\WEZ.tmp")
Call BvpEnv("PACMM3","PAC7GG",Rep_TMP & "\WGG.tmp")
Call BvpEnv("PACMM3","PAC7JG",Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM3","PAC7KC",Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM3","PAC7KG",Rep_TMP & "\WKG.tmp")
Call BvpEnv("PACMM3","PAC7G1",Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM3","PAC7G2",Rep_USR & "\COBOL_MIA3.txt")
Call BvpEnv("PACMM3","PAC7G3",Rep_USR & _
"\CONTROL_COBOL_MIA3.txt")
Call BvpEnv("PACMM3","PAC7W1",Rep_TMP & "\WW1.tmp")
Call BvpEnv("PACMM3","PAC7W2",Rep_TMP & "\WW2.tmp")
Call BvpEnv("PACMM3","PACM2P",Rep_TMP & "\W2P.tmp")
Call BvpEnv("PACMM3","PACM2W",Rep_TMP & "\W2W.tmp")

```

```

Call BvpEnv("PACMM3","PAC2GP",Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM3","PAC2GW",Rep_TMP & "\WGW.tmp")
Call BvpEnv("PACMM3","PAC780",Rep_TMP & "\W80.tmp")
Call BvpEnv("PACMM3","PAC7IA",Rep_USR & "\MIA3IA.txt")
Call RunCmdLog ("BVPACMM3")
Call Err_Cod(Return , 0 , "PACMM3")

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

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

</script>
</job>

```

MIA4 - Server Generation

MIA4 - Introduction

This procedure enables you to generate the Servers in two files: a file that contains the generated COBOL code and another file that contains the control COBOL code.

It generates the micropatterns which do not come from Macros.

The generation is carried out without line numbers.

The right part (columns 73 to 80) is generated as a blank, except for the function tag lines where lvnn is generated (with nn = level of the function/subfunction).

MIA4 - User Input

A '*' line with the user code, password and Library code is required.

Pos.	Lon.	Value	Meaning
2	1	*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code

You must then specify a GGS command line as described in the input of the GPRT procedure for each server to be generated.

MIA4 - Description of Steps

Generation/Print commands : BVPACA10

Code	Physical name	Type	Label
PAC7AR	DBase dir.: AR	Input	Development Database data
PAC7AN	DBase dir.: AN	Input	Development Database index
PAC7AY	DBase dir.: AY	Input	Development Database extension
PAC7AJ	DBase dir.: AJ	Input	Development Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir.: GU	Input	Administration Database users
PAC7ME	User Input	Input	User transactions
PAC7MV	Tmp dir.: WMV	Output	Update transactions
PAC7MG	Tmp dir.: WMG	Output	Generation/Print commands

Dispatch of command lines : BVPACA20

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PACGGK	Admin. Database - DBase dir.: GK	Input/ Output	Generation rights
PAC7JG	Tmp dir.: WJG	Output	Extractor command
PAC7KF	Tmp dir.: WKF	Output	Work file
PAC7MG	Tmp dir.: WMG	Input	Generation/Print commands
PAC7W1	NUL	Input/ Output	Work file

Code	Physical name	Type	Label
PAC7KU	Tmp dir.: WKU	Output	Output report

Server generation : BVPACMM4

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - DBase dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - DBase dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - DBase dir.: GU	Input	Administration Database users
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AY	Database dir.: AN	Input	Development Database extension data
PAC7SS	System - Skel. dir.: SS	Input	Generation skeleton
PAC7EV	Tmp dir.: WEV	Input	Work file
PAC7GV	Tmp dir.: WGV	Output	Work file
PAC7JG	Tmp dir.: WJG	Output	Work file
PAC7KC	Tmp dir.: WKC	Output	Work file
PAC7KV	Tmp dir.: WKV	Output	Work file
PAC7G1	Tmp dir.: WG1	Output	Work file
PAC7G2	User dir.: COBOL_MIA4	Output	C9 COBOL file
PAC7G3	User dir.: CONTROL_COBOL_MIA4	Output	Control COBOL file
PAC7W1	Tmp dir.: WW1	Output	Work file
PAC7W2	Tmp dir.: WW2	Output	Work file
PAC7EZ	Tmp dir.: WEZ	Output	Work file
PACM2W	Tmp dir.: W2W	Output	Work file
PACM2P	Tmp dir.: W2P	Output	Work file
PAC2GW	Tmp dir.: WGW	Output	Work file
PAC2GP	Tmp dir.: WGP	Output	Work file
PAC780	Tmp dir.: W80	Output	Work file

Code	Physical name	Type	Label
PAC7IA	User dir.: MIA2IA	Report	General printout of the program stream

MIA4 - Execution Script

```

| -----
|          VISUALAGE PACBASE
| -----
|          SERVER GENERATION
| -----
|
<job id=MIA4>

<script language="VBScript">
Dim MyProc
MyProc = "MIA4"
</script>

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

<script language="VBScript">

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

Dim CodLang
If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
Else
CodLang = WshShell.RegRead (Rep_SYS & "\GENLANG")
End If

Call Msg_Log (Array("1022" , "PACA10"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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("PAC7ME") = Fic_Input
Call BvpEnv("PACA10","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PACA10","PAC7MG",Rep_TMP & "\WMG.tmp")
Call RunCmdLog ("BVPACA10")
Call Err_Cod(Return , 0 , "PACA10")

Call Msg_Log (Array("1022" , "PACA20"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"

```

```

WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGK") = Rep_ABASE & "\GK"
Call BvpEnv("PACA20", "PAC7JG", Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACA20", "PAC7KU", Rep_TMP & "\WКУ.tmp")
Call BvpEnv("PACA20", "PAC7KF", Rep_TMP & "\WKF.tmp")
Call BvpEnv("PACA20", "PAC7MG", Rep_TMP & "\WMG.tmp")
Call BvpEnv("PACA20", "PAC7W1", Rep_TMP & "\NUL.tmp")
Call RunCmdLog ("BVPACA20")
Call Err_Cod(Return , 0 , "PACA20")

Call Msg_Log (Array("1022" , "PACMM4"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7SS") = Rep_SKEL & "\SS" & CodLang
Call BvpEnv("PACMM4", "PAC7EV", Rep_TMP & "\WEV.tmp")
Call BvpEnv("PACMM4", "PAC7EZ", Rep_TMP & "\WEZ.tmp")
Call BvpEnv("PACMM4", "PAC7GV", Rep_TMP & "\WGV.tmp")
Call BvpEnv("PACMM4", "PAC7JG", Rep_TMP & "\WJG.tmp")
Call BvpEnv("PACMM4", "PAC7KC", Rep_TMP & "\WKC.tmp")
Call BvpEnv("PACMM4", "PAC7KV", Rep_TMP & "\WKV.tmp")
Call BvpEnv("PACMM4", "PAC7G1", Rep_TMP & "\WG1.tmp")
Call BvpEnv("PACMM4", "PAC7G2", Rep_USR & "\COBOL_MIA4.txt")
Call BvpEnv("PACMM4", "PAC7G3", Rep_USR & _
"CONTROL_COBOL_MIA4.txt")
Call BvpEnv("PACMM4", "PAC7W1", Rep_TMP & "\W1.tmp")
Call BvpEnv("PACMM4", "PAC7W2", Rep_TMP & "\W2.tmp")
Call BvpEnv("PACMM4", "PACM2P", Rep_TMP & "\W2P.tmp")
Call BvpEnv("PACMM4", "PACM2W", Rep_TMP & "\W2W.tmp")
Call BvpEnv("PACMM4", "PAC2GP", Rep_TMP & "\WGP.tmp")
Call BvpEnv("PACMM4", "PAC2GW", Rep_TMP & "\WGW.tmp")
Call BvpEnv("PACMM4", "PAC780", Rep_TMP & "\W80.tmp")
Call BvpEnv("PACMM4", "PAC7IA", Rep_USR & "\MIA4IA.txt")
Call RunCmdLog ("BVPACMM4")
Call Err_Cod(Return , 0 , "PACMM4")

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

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

</script>
</job>

```

TRDQ - Conversion of DBD Blocks into SQL Blocks

TRDQ - Introduction

Principle

The TRDQ utility transforms DB2 Database Blocks (DB type) into SQL DBD Blocks (Q2 type).

First step

From the command lines, input lines of RMEN are formatted with the RENAME option (RN).

Second step

RMEN renames the DB-type Block codes into new Block codes that are specified in the command line.

RMEN creates a sequential file that contains:

- Creation and modification transactions with the new Block code, sorted by:
 - ascending library hierarchical level,
 - library,
 - record type (definitions, descriptions, uses).
- Deletion transactions with the old Block code, sorted by:
 - descending library hierarchical level,
 - library,
 - record type (uses, descriptions, definitions).

Third step

Transformation of DB-type Blocks into Q2-type Blocks.

From the output file of RMEN and the command lines, transformation of records:

- Definition of new Blocks: the DB-type is transformed into Q2-type. The external name and the label can be modified.
- Description of Codasyl-DB2 Database Blocks (-DC): it is transformed into the description of relational Blocks (-DR).
- Documentation of Codasyl-DB2 Database Blocks description (-DC...GG; -DC...GC): it is transformed into the documentation of relational Blocks description (-DR...GG; -DR...GC).

The other records produced by RMEN are copied without transformation in the output transactions file.

Execution conditions

None, since the database is not directly updated.

Batch procedure access authorization option: the use of the RMEN procedure is subject to a separate purchase agreement.

Printed output

This procedure prints, in the PACX (RMEN) step, a selection report with the errors, if any.

Result

The output file is a sequential file that contains the transformed Q2 Block transactions. These transactions will be used as input to the Database Update (UPDT) procedure.

Note

It is possible to keep the original Block code by running RMEN to rename the new Q2-type code.

TRDQ - User Input

One '*' line containing the DB-type DBD Block(s) to be transformed:

This constraint is imposed by RMEN.

The procedure must be executed for each new extraction context:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Library code to extract
22	4	ssss	Session number (default=current)
26	1	'T'	Test version of frozen session
29	4	RMEN	Code of extractor
33	1	'1'	Formatting for UPDT
50	1	' '	Password not transferred into the output file

Pos.	Len.	Value	Meaning
		'1'	The password is transferred

One command line per DB-type DBD Block to be transformed into Q2-type SQL Block:

Pos.	Len.	Value	Meaning
2	2	'W1'	Line code
4	6	bbbbbb	Pacbase code of the DB-type Block to be transformed
10	6	qqqqqq	Pacbase code of the resulting Q2-type Block. This new code must be different from any existing Block code.
			Optional modifications for the resulting Q2-type Block:
16	8	eeeeeeee	External name of the Q2-type Block
24	1	x	Control card in front of the Q2 Block
25	1	y	Control card in back of the Q2 Block
26	36	nn....	Q2 Block name
62	2	Q2	Q2 Block type

TRDQ - Description of Steps

RMEN input formatting: BVPTUDQ1

Code	Physical name	Type	Label
PAC7MB	User input	Input	User input
PAC7ME	Tmp dir.: WMB	Output	Input file for RMEN

Extraction: PACX

This step extracts transactions according to user input.

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

Code	Physical name	Type	Label
PACGGR	Admin Database - DBase dir. : AR	Input	Administration Database data
PACGGU	Admin Database - DBase dir. : GU	Input	Administration Database users
PAC7MB	User Input	Input	User input
PAC7MA	NUL	Input	Work file
PAC7ES	NUL	Input	Work file
PAC7BM	Tmp dir.: WBM	Input/Output	User input
PAC7MM	Tmp dir.: WMM	Input/Output	Work file
PAC7MJ	Tmp dir.: WMJ	Input/Output	Work file
PAC7TE	Tmp dir.: WTE	Input/Output	Work file
PAC7RE	Tmp dir.: WRE	Input/Output	Work file
PAC7RM	Tmp dir.: WRM	Input/Output	Work file
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted transactions
SYSEXT	Tmp dir.: WSY	Input/Output	Work file
PAC7MV	Tmp dir.: WMV	Output	Work file
PAC7CP	NULCP	Output	Work file
PAC7MR	Tmp dir.: WMR	Output	Work file
PAC7MX	Tmp dir.: WMX	Output	Entities not extracted (PACX)
PAC7GY	NUL	Output	Extracted transactions for UPDP
PAC7RQ	Tmp dir.: WQR	Output	Matching Entity/Library
PAC7TD	Tmp dir.: WTD	Output	Work file
PAC7UE	Tmp dir.: WUE	Output	Work file
PAC7IA	User dir.: PACXIA	Report	General printout of the program stream
PAC7DD	User dir.: PACXDD	Report	Errors on input transactions
PAC7ED	Tmp dir.: WED	Report	Extractions report
PAC7EE	User dir.: PACXEE	Report	Extractions report
PAC7EG	Tmp dir.: WEG	Report	Extractions report
PAC7EM	Tmp dir.: WEM	Report	Extractions report
PAC7EP	User dir.: PACXEP	Report	Extractions report
PAC7EQ	User dir.: PACXEQ	Report	Extractions report
PAC7EU	Tmp dir.: WEU	Report	Extractions report

Code	Physical name	Type	Label
PAC7EZ	User dir.: PACXEZ	Report	Extractions report

Transformation of DB Blocks into SQL Blocks: BVPTUDQ3

Code	Physical name	Type	Label
PAC7MB	User input	Input	User input
PAC7MX	Tmp dir.: WMV	Input	File produced by RMEN
PAC7MV	User dir.: UPDATESQL	Output	Update transactions

TRDQ - Execution JCL

```

| -----
|          VISUALAGE PACBASE
|
| -----
|
|          CONVERSION OF DBD BLOCKS INTO SQL BLOCKS
|
| -----
|
<job id=TRDQ>

<script language="VBScript">
Dim MyProc
MyProc = "TRDQ"
</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" , "PTUDQ1"))
'-----
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTUDQ1","PAC7ME",Rep_TMP & "\WMB.tmp")
Call RunCmdLog ("BVPTUDQ1")
Call Err_Cod(Return , 0 , "PTUDQ1")

Call Msg_Log (Array("1022" , "PACX"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
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"

```

```

Call BvpEnv("PACX","PAC7MB",Rep_TMP & "\\WMB.tmp")
Call BvpEnv("PACX","PAC7BM",Rep_TMP & "\\WBM.tmp")
Call BvpEnv("PACX","PAC7WD",Rep_TMP & "\\WWD.tmp")
Call BvpEnv("PACX","PAC7MM",Rep_TMP & "\\WMM.tmp")
Call BvpEnv("PACX","PAC7MJ",Rep_TMP & "\\WMJ.tmp")
Call BvpEnv("PACX","PAC7TE",Rep_TMP & "\\WTE.tmp")
Call BvpEnv("PACX","PAC7RE",Rep_TMP & "\\WRE.tmp")
Call BvpEnv("PACX","PAC7RQ",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7RM",Rep_TMP & "\\WRM.tmp")
Call BvpEnv("PACX","PAC7MA",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7ES",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7UE",Rep_TMP & "\\WUE.tmp")
Call BvpEnv("PACX","PAC7GY",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7CP",Rep_TMP & "\\NUL.tmp")
Call BvpEnv("PACX","PAC7TD",Rep_TMP & "\\WTD.tmp")
Call BvpEnv("PACX","PAC7MV",Rep_TMP & "\\WMV.tmp")
Call BvpEnv("PACX","PAC7MR",Rep_TMP & "\\WMR.tmp")
Call BvpEnv("PACX","PAC7MX",Rep_TMP & "\\WMX.tmp")
Call BvpEnv("PACX","PAC7IA",Rep_USR & "\\PACXIA.txt")
Call BvpEnv("PACX","PAC7DD",Rep_USR & "\\PACXDD.txt")
Call BvpEnv("PACX","PAC7ED",Rep_TMP & "\\WED.tmp")
Call BvpEnv("PACX","PAC7EE",Rep_USR & "\\PACXEE.txt")
Call BvpEnv("PACX","PAC7EG",Rep_TMP & "\\WEG.tmp")
Call BvpEnv("PACX","PAC7EM",Rep_TMP & "\\WEM.tmp")
Call BvpEnv("PACX","PAC7EP",Rep_USR & "\\PACXEP.txt")
Call BvpEnv("PACX","PAC7EQ",Rep_USR & "\\PACXEQ.txt")
Call BvpEnv("PACX","PAC7EU",Rep_TMP & "\\WEU.tmp")
Call BvpEnv("PACX","PAC7EZ",Rep_USR & "\\PACXEZ.txt")
Call BvpEnv("PACX","SYSEXT",Rep_TMP & "\\WSY.tmp")
Call RunCmdLog ("BVPACX")
If Return = 4 Then
Call Msg_Log (Array("1030"))
End If
If Return = 8 Then
Call Msg_Log (Array("1057"))
End If
Call Err_Cod(Return , 4 , "PACX")

Call Msg_Log (Array("1022" , "PTUDQ3"))
'-----
WshEnv("PAC7MB") = Fic Input
Call BvpEnv("PTUDQ3","PAC7MV",Rep_USR & "\\UPDATESQL.txt")
Call BvpEnv("PTUDQ3","PAC7MX",Rep_TMP & "\\WMV.tmp")
Call RunCmdLog ("BVPTUDQ3")
Call Err_Cod(Return , 0 , "PTUDQ3")

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

Call Msg_Log (Array("1023"))
'-----

```

```
Wscript.Quit (Return)
```

```
</script>
```

```
</job>
```

Chapter 9. Components

Server Environment Components

Introduction

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

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

On-Line Documentation

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

Generation Skeletons

The product also includes the following files:

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

Administration Database

Administration Database Files

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

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

- Journal file of SCM Tools Interface, QJ

Administration Database Backup

The Administration Database backup consists of two sequential generation files.

- Backup of the Database (PC).

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

- Backup of the journal (PJ).

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

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

- Backup of SCM module QJ journal

The backup of QJ is the JQ file.

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

Development Database

Development Database Files

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

- The Data file (AR).

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

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

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

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

Development Database Backup Files

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

- Database or Data backup (PC).

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

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

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

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

Modules - Specific Files

Pac/Impact:

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

DSMS:

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

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

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

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

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

PAF:

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

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

- Work file for PAF in batch mode

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

PAF extension

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

Complementary Libraries and Files

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

PQC function

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

- Specific members of Pacbench Quality Control function

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

MIAM procedure.

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

Member	Content or format	Remarks
BVPAMIAM	Sequential file	Standard parameters for the procedure MIAM

Chapter 10. Appendix

Installation of the Administration Database Model

VINS - Introduction

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

Execution conditions

The Database must be closed to on-line processing.

Abnormal execution

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

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

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

VINS - Input / Processing / Results

This procedure requires two types of user input:

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

The structure of the line is the following:

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

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

Printed output

The procedure outputs:

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

Result

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

Note

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

VINS - Description of Steps

Update of the Administration Database: VINS

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

Code	Physical name	Type	Label
PAC7AY	Admin. Database - Base dir.: AY	Output	Administration Database Extension
PAC7MA	System - skel. dir.: BVPMETAD	Input	IBM Meta Entities Transactions
PAC7MB	User input	Input	User Input
PAC7BM	Tmp dir.: WBM	Input/ Output	Work file
PAC7WD	Tmp dir.: WWD	Input/ Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/ Output	Extracted Transactions
PAC7TD	User dir.: TD	Input/ Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR on Administration Base
PAC7MX	NUL	Output	Non extracted entities
PAC7RQ	NUL	Output	Work file
PAC7IA	User dir.: VINSIA	Report	Complete printing of programs sequence
PAC7CP	NULCP		
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 errors and restart the procedure
- 8 : No access authorization for batch procedure
- 12: Input-output error on a file

VINS - Execution Script

```
'-----  
'          VISUALAGE PACBASE  
'-----  
'          - DICTIONARY UPDATING WITH IBM MODEL DEVPT -  
'-----  
'  
' THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE  
' DATABASE, BASED ON TRANSACTIONS PROVIDED.  
'  
' INPUT :  
' - USER IDENTIFICATION LINE (REQUIRED)  
'   COL 2 : "*"   
'   COL 3 : USERIDXX  
'   COL 11 : PASSWORD  
'   COL 29 : "VINS"  
'   COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES  
'           "R" - RETRIEVAL OF USER ENTITIES WITH THE  
'             "EXTENSION" FORMAT  
'           " " "I" + "R"  
'-----  
  
<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  
  
' No Rollback  
WshEnv("BVPRB") = "N"  
  
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 & "\PAC7xx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS", "PAC7TD", Rep_USR & "\TD.txt")

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS", "PAC7MR", Rep_TMP & "\WMR.tmp")
Call BvpEnv("VINS", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("VINS", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("VINS", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS", "PAC7CP", Rep_TMP & "\NULCP.tmp")
Call BvpEnv("VINS", "SYSEXT", Rep_TMP & "\VINSYS.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

Position	Length	Value	Meaning
		'R'	Retrieval of User Entities with the 'extension' format
		' '	'T' + 'R'

Printed output

The procedure prints

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

Result

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

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

Note

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

VINS - Description of Steps

Update of the Development Database : VINS

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error Labels
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database Index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database Data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database Extension

Code	Physical name	Type	Label
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database Users
PAC7AJ	Journal dir.: AJ	Output	Development Database Journal
PAC7AN	Base dir.: AN	Output	Development Database Index
PAC7AR	Base dir.: AR	Output	Development Database Data
PAC7AY	Base dir.: AY	Output	Development Database Extension
PAC7MA	System - skel. dir.: BVPMETBA	Input	IBM Meta Entities Transactions
PAC7MB	User input	Input	User Input
PAC7BM	Tmp dir.: WBM	Input/ Output	User input
PAC7WD	Tmp dir.: WWD	Input/ Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/ Output	Extracted Transactions
PAC7TD	User dir.: TD	Input/ Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR
PAC7MX	NUL	Output	Non extracted entities
PAC7RQ	NUL	Output	Work file
PAC7IA	User dir.: VINSIA	Report	Complete printing of programs sequence
PAC7CP	NULCP		
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 errors and restart the procedure
- 8 : No access authorization for batch procedure
- 12: Input-output error on a file

VINS - Execution Script

```

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

' No Rollback
WshEnv("BVPRB") = "N"

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 & "\NUL EE.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", "PAC7RQ", Rep_TMP & "\NUL.tmp")
Call BvpEnv("VINS", "PAC7WD", Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS", "PAC7CP", Rep_TMP & "\NULCP.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.

Execution condition

On-line servers must be shut down.

User input

None.

LDGS - Description of Steps

Initialization of the GS file: PTLDGS

This step initializes the GS file with a record.

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

LDGS - Execution Script

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

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

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

```

<script language="VBScript">

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

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: PTL DHE

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

```
'-----  
'      VISUALAGE PACBASE  
'-----  
<job id=LDHE>  
  
<script language="VBScript">  
MyProc = "LDHE"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
Call Msg_Log (Array("1029" ))  
'-----  
Call StateList (base, statusL)  
  
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

```
'-----  
'      VISUALAGE PACBASE  
'-----  
<job id=LDZS>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "LDZS"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
Call Msg_Log (Array("1029" ))  
'-----  
Call StateList (base, statusL)  
  
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 the programs' 'P' lines which contain the 'UNS' operator and do not contain anything in the 'Level-Condition type' field. The user will have to check the output file and delete, in this file, all the lines he/she wants to keep as they are. For the other lines, the 'Level-Condition type' field will have to be forced to '99BL' before the execution of the UTU2 procedure.

Execution conditions

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

Result

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

UTU1 - Input / Processing / Results

No user input.

UTU1 - Description of Steps

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

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

UTU1 - Execution Script

```
|-----|
| VISUALAGE PACBASE |
|-----|
| EXTRACTION OF LINES "P" |
| WITH OPERATOR "UNS" |
| WITHOUT LEVEL-CONDITION TYPE |
```

```

' -----
'
<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","UTUT0",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 forced to '99BL'.

Execution conditions

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

Result

A new sequential image of the Development Database.

UTU2 - Input / Processing / Results

No user input.

UTU2 - Description of Steps

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

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

UTU2 - Execution Script

```
| -----  
|          VISUALAGE PACBASE  
| -----  
|          UPDATE LINES "P"  
|          WITH OPERATOR "UNS"  
| -----  
|  
<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 whose purpose is to convert the 'old' meta entities into 'formatted' meta entities.

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

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

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

Migration process

Step 1: Analysis of the Database relations (UTM1)

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

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

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

The list of calls is printed in the output file.

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

You will have to specify the new relation code.

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

Example of transformation:

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

modified as:

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

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

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

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

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

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

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

- The relations you have selected are modified or created.

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

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

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

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

- Relations calls are updated in the concerned entities (description lines of meta entities and others).

- If errors are detected (return code 8), you will have to modify the input file PAC7ME and start again the UTM2 procedure before the update. Caution, the &UTM2MV permanent file must first be deleted or renamed.

Update

The update is performed by the BVPACA15 program.

A backup of the Database is recommended before the update.

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

The transactions are not journalized.

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

Step 3: Inventory of fixtures

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

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

Step 4: Reorganization

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

Execution conditions

None during step 1 (UTM1).

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

Printed output

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

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

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

Result

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

UTM1 - Description of Steps

Analysis of the Database relations: PTUME1

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Development Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Development Database Index
PAC7ME	User dir. : UTM1ME	Output	List of relations calls in all the entities (length = 221)
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7EQ	User dir.: UTM1EQ	Report	Report

UTM1 - Execution Script

```

| -----
|          VISUALAGE PACBASE
| -----
|          CONVERSION OF OLD META-ENTITIES
|          ANALYSIS OF THE DATABASE RELATIONS
| -----
|
<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 as input to the second step of the conversion (UTM2) with a user code and its password.

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

UTM2 - Description of Steps

Consistency check of the relations file : PTUME3

Code	Physical name	Type	Label
PAC7ME	User dir.: UTM1ME	Input	List of relation calls in all the entities

Conversion of the meta entities, the relations and their calls: PTUME2

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Development Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Development Database index
PAC7ME	User dir. : UTM1ME	Input	List of relations calls in all the entities
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7MV	User dir.: UTM2MV	Output	Update transactions (length = 170)
PAC7ET	User dir.: UTM2ET	Output	Report
PAC7MB	User input	Input	user input

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Output	Development Database data
PAC7AN	Database dir.: AN	Output	Development Database index
PAC7AY	Database dir.: AY	Output	Development Database extension
PAC7AJ	NUL	Output	Development Database Journal
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7DC	NUL	Input	DSMS elements file of the Development Database
PAC7ME	NUL	Input	Work file
PAC7MV	User dir.: UTM2MV	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 transactions belonging to a user is preceded by a banner specifying the user code.

Return codes:

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

UTM2 - Execution Script

```

| -----
|          VISUALAGE PACBASE
| -----

```

```

'           CONVERSION OF OLD META-ENTITIES
'           THE RELATIONS AND THEIR CALLS
' -----
'
<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

Call Msg_Log (Array("1022" , "PTUME3"))
' -----
Call BvpEnv("PTUME3","PAC7ME",RepT_USR & "\UTM1ME.TXT")
Call RunCmdLog ("BVPTUME3")
Call Err_Cod(Return , 0 , "PTUME3")

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("SEADMIN") = 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 , "ACA15")

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

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

</script>
</job>

```




Part Number: DELNT001367A - 9337

Printed in USA