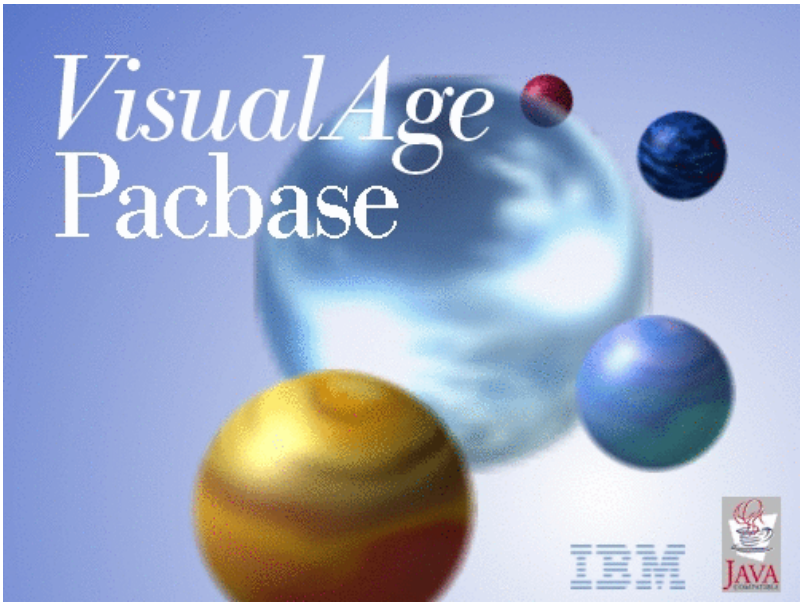
VisualAge Pacbase

Convergence plan: Installation and Procedures
OS/390 CICS

Version 3.5



VisualAge Pacbase

Convergence plan: Installation and Procedures OS/390 CICS

Version 3.5

Note

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

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

First Edition (February 2009)

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 France Software Laboratory, Rational Division
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.

Contents

Notices

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM France Software Laboratory - Rational Division, 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. Installation

Setting

Preparation

The preparation consists of three stages:

- Receiving the product with the SMP/E (System Modification Program/Extended) utility in dedicated PDSs.

  This stage must be performed by the system staff who are accustomed to installing IBM products with SMP/E.
- Allocating a PDS file where all the installation and operation JCLs will be saved.
- Executing the installation JCL from the PDS members created during the first stage.

Important note:

This installation is dedicated to implementing components which are specific to the migration procedures.

This implies that the SMP/E environment and the installation targets of any kind must be entirely distinct from the VA Pac installation.

Stage 1

The product is delivered on a CD-ROM.

To install it, you must:

1 - Download to Z/OS the sample JCLs provided on the CD-ROM.

On the CD-ROM, a file contains the sample JCLs: a JCL for the allocation of the compressed RELFILEs and SMPMCS files, as well as the SMP/E configuration and installation JCLs.

1.1 - Allocate first the receiving file on Z/OS, either by parameterizing the JCL presented here below, or by directly creating it with all the required characteristics (tsouid represents the TSO user code):

```
//ALLOC1   EXEC PGM=IEFBR14
//*
//FTPALLO  DD DSN=tsouid.HBVP950.SAMPLE.JCL.BIN,
```

```
//         DISP=(NEW,CATLG,DELETE),
//         DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160,
//         SPACE=(TRK,(2,1)),UNIT=SYSALLDA
//*        VOL=SER=&TVOL1
```

1.2 - Download the binary-formatted files provided on the CD-ROM to the Z/OS file already created.

To do so, perform an FTP transfer as follows:

1.2.1 - Copy the FTP sample command 'ftpini_model.scr' provided on the CD-ROM to a working directory of your workstation and name it 'ftpini.scr' ; then follow the setting instructions contained in this file.

1.2.2 - Execute the FTP command -s:ftpini.scr from the working directory.

2 - Extract the sample JCLs with the TSO command RECEIVE INDA('tsouid.HBVP950.SAMPLE.JCL.BIN).

This creates the tsouid.HBVP950.TMP.JCLINST file with the following members:

- BVPALLOC
- BVPMACC
- BVPMAPP
- BVPMREC
- BVPXMIT
- BVP1DCSI
- BVP2ICSI
- BVP3ALLO
- BVP4DEFZ
- BVP5DDEF
- BVP6DDEF
- BVP7ALLO

3 - Download the compressed RELFILEs and the SMPMCS from the CD-ROM.

3.1 - Parameterize and submit the BVPALLOC JCL contained in tsouid.HBVP950.TMP.JCLINST in order to allocate, on Z/OS, the files required for the download operations.

To do so, follow the instructions included in the JCL.

3.2 - Download the binary-formatted files provided on the CD-ROM to the Z/OS files.

To do so, perform an FTP transfer as follows:

3.2.1 - Copy the FTP sample command 'ftpput_model.scr' provided on the CD-ROM to a working directory of your workstation and name it 'ftpput.scr' ; then follow the setting instructions contained in this file.

3.2.2 - Execute the FTP command -s:ftpput.scr from the working directory.

4 - Extract the RELFILEs files on Z/OS.

To do so, parameterize and submit the BVPXMIT JCL contained in tsouid.HBVP950.TMP.JCLINST

5 - From then on, you can install the product with SMP/E, using the JCLs contained in the tsouid.HBVP950.TMP.JCLINST PDS.

If a previous installation exists, you must follow the instructions indicated in the VA Pac Installation Manual, chapter 'Installation of Server Components', section 'Preparation', paragraph 'Remark in case of a previous SMP/E installation'.

If you install in a new context, you will have to create the SMP/E environment and install the FMID HBVP950, using the JCLs supplied.

This stage then consists of the following steps:

- Setting up the SMP/E environment and the 'Target zone' and 'Distribution zone' files of the root FMID HBVP950:
  - Defining the SMP/E cluster libraries (BVP1DCSI)
  - Initializing the SMP/E libraries (BVP2ICSI)
  - Allocating the SMP/E work files (BVP3ALLO)
  - Assigning the FMID (BVP4DEFZ)
  - Creating the DDDEF entries for technical files (BVP5DDEF)
  - Defining the 'Target zone' and 'Distribution zone' files (BVP6DDEF)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP7ALLO)
- Installing the components of the FMID HBVP950.
  - Execution of RECEIVE (BVPMREC)
  - Execution of APPLY (BVPMAPP)
  - Execution of ACCEPT (BVPMACC)

At the end of this first stage, all the components required for the installation are located in the following PDSs ; 'hlq' indicates the common prefix of the elements supplied (High-Level Qualifier):

- hlq.SBVPINST: all the 80-long files:
  - The error messages file (BVPAE)
  - The initial installation JCL (BVPMINIT)
  - A file which contains the installation and operation JCLs and procedures (BVPMTTAL)
  - The SMP/E sample JCLs
- hlq.SBVPMBR8: batch load-modules
- hlq.SBVPMF4: SC skeleton

Stage 2

This stage is optional but recommended. It consists in allocating a PDS file with the following characteristics:

- Lrecl=80
- Size: around 100 tracks of a 3390 disk, 30-pad directory.

Stage 3

Stage 3 consists in copying the 'hlq.SBVPINST(BVPMINIT)' JCL to the PDS mentioned in stage 2, modifying its parameters to match the constraints of the site and executing it to obtain the complete installation and operation JCL.

The BVPMINIT JCL executes the BVPMMJCL program loaded into the hlq.SBVPMBR8 PDS.

It must be completed as follows:

- Fill in '&hlq' with the value of the 'hlq' parameter used in the SMP/E first stage.
- In the '//SYSUT2 DD DSN=' field, enter the name of the file into which the complete installation-operation JCL is to be loaded.

  This file can be either a PDS member initially created to receive all the JCLs, or a sequential file selected by the user.
- Enter the parameters (see details in the next chapter).

The BVPMMJCL program execution must be saved: it can be used for re-installations.

Execution JCL

```
//VAPACMSM  JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//MM1JCL EXEC PGM=BVPMMJCL
//STEPLIB  DD DISP=SHR,DSN=&HLQ.SBVPMBR8
//         DD DISP=SHR,DSN=---.---.---    LE LIBRARY
//SYSOUT   DD SYSOUT=A
//SYSUT1   DD DSN=&HLQ.SBVPINST(BVPMTTAL),DISP=SHR
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT8   DD DUMMY,DCB=BLKSIZE=1370
//SYSUT9   DD DUMMY,DCB=BLKSIZE=1370
//*************************************************************
//*
//*     CREATION OF INSTALLATION JCL THROUGH 'BVPMMJCL'
//*     -----------------------------------------------
//*
//*     MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ASKING,
//*     IF NECESSARY, A SELECTION OF PARTS OF INSTALLATION JCL
//*     (JCL MODULES), BY GIVING THE APPROPRIATE VALUES TO THE
//*     INSTALLATION PARAMETERS, AND, IF NECESSARY, BY SPECIFYING
//*     THE LINES TO BE ADDED AT THE BEGINNING OR AT THE END OF
//*     EACH JCL MODULE.
//*************************************************************
//SYSPRM DD DUMMY
//SYSUT2   DD ---------   PDS MEMBER OR SEQUENTIAL FILE RECEIVING
//*                       THE INSTALLATION JCL (LRECL=80)
//SYSIN DD *
===PRM PRFJ=PAC           .JOB NAMES PREFIXES (3 CHARACTERS)
===PRM CCPT=<>            .JOB ACCOUNTING CODES (JOB CARDS)
===PRM CLASSJ=1           .JOB EXECUTION CLASS (JOB CARDS)
===PRM MSGCL=A            .JCL OUTPUT CLASS (MSGCLASS)
===PRM BASE='BASE'        .NAME OF DATABASE DEVELOPMENT (4 CHAR)
===PRM OUT=A              .JOB SYSOUT CLASS
===PRM INDSV='EXP.BVPPV'  .SYSTEM VSAM FILES VA-PAC
===PRM INDUV='UTI.BVPPV'  .USER VSAM FILES VA-PAC
===PRM INDMV='MIG.BVPMV'  .MIGR VSAM FILES
===PRM INDMN='MIG.BVPMN'  .MIGR NON VSAM FILES
===PRM VOLUN=             .USER NON VSAM FILES VOL=SER
===PRM VOLMN=             .MIGR NON VSAM FILES VOL=SER
===PRM VOLMV=             .MIGR VSAM FILES VOLUME
===PRM VCAT=<>            .CATALOG OF DSMS USER VSAM FILES
===PRM SCAT=<>            .CATALOG OF DSMS SYSTEM VSAM FILES
===PRM LSK='A'            .LANGAGE OF SKELETON FILES
===PRM UWK=SYSDA          .WORK UNIT
===PRM UNITUN=SYSDA       .NON VSAM USER FILES UNIT
===PRM UNITMN=SYSDA       .NON VSAM MIGRATION FILES UNIT
===PRM HLQ='HLQ'          .HIGH LEVEL QUALIFIER OF LOAD MODULES
===PRM BIBP='SYS1.PROCLIB' .PROCEDURE LIBRARY
===PRM BIBT='SYS1.SORTLIB' .SORT LIBRARY
===PRM BCOB='SYS1.SCEERUN'     .COBOL ROUTINE LIBRARY
===BEGMOD
./ ADD NAME=$ZMODUL
/*
//
```

JCL installation

The BVPMMJCL module reads the skeleton JCL file and produces an appropriate JCL, with commands which enable you to:

- Select portions of the skeleton JCL, which are called 'JCL modules',
- Parameterize the skeleton in order to obtain a JCL which requires a minimum of modifications to be operational,
- Add lines before and/or after the JCL modules to separate each one.

This step can be executed as many times as necessary to generate an appropriate JCL.

USER INPUT

Command	Parameters	Comments
===PRM	PPPP=pppp (1)	Parameter
===SELM	jcl1 jcl2	Selected JCL modules
===BEGMOD		Insertion of lines at the beginning of module
....1		Lines to be added before each module
....n		
===ENDMOD		Insertion of lines at the end of module
....1		Lines to be added after each module
....n		

(1) PPPP = parameter name, pppp = parameter value

Notes

- Lines ===PRM

  You can add a comment but it must be preceded by a dot and it should not exceed column 72.

  The default values are only examples. You must therefore enter values adapted to your site.

- Lines ===SELM

  These lines can be used to select modules.

  Since the standard installation provides all the modules, this line is not used.

- Lines ===BEGMOD

  ./ ADD NAME=$ZMODUL

  As a result, a line is inserted before each JCL module, in the form:

./ ADD NAME=<JCL-module>

RESULT: the complete JCL

The file obtained in SYSUT2 contains all the installation and operation JCLs. This file must be open with an editor to launch the installation process.

Two operations must be performed on the complete JCL:

1. Overall modifications (if necessary)

Adaptations can be performed on all the JCLs.

The VSAM catalogs appear as comments in the JCL obtained after the installation:

- In the DELETE/DEFINE*/
- In the JCL STEPCATs
- In the declarations of the procedures' parameters

When these parameters are not required on the site, the resulting JCL can remain as it is.

When these parameters are required on the site, the relevant lines should be changed into command lines. To do so, you must:

- Transform all '//*:' into '//',
- And then replace '/*:' and '*/' with blanks.

Caution: SMS

- In the installation jobs which include the 'GenerationDataGroup' allocation, you must delete the lines DD //GDGMOD from the definition IDCAMS.
- If the UNIT and VOL parameters cannot be used on the site, you can delete them in the whole JCL via an exclude command (EXCLUDE command in TSO/EDIT).

It is usually recommended to perform any other overall modifications on the JCLs before splitting these JCLs.

Caution: LSR

JCLs are standardly provided with the LSR option for the optimization of the batch access to VSAM files.

If the LSR option is not implemented on the site, you must replace the following lines in the procedures:

```
//xxLSR DD DSN=&INDxx..file,DISP=SHR
//PACxx DD SUBSYS=(&LSR,'DDNAME=xxLSR','BUFND=10','BUFNI=10
```

with a line:

```
//PACxx DD DSN=&INDxx..file,DISP=SHR.
```

2. JCL splitting

Before each module of a standard complete JCL, there is a ./ ADD NAME=<JCL-module> line, where <JCL-module> is the code of the ===MOD line found (see the following table of JCL modules).

This allows the complete JCL to be split into as many members as JCL modules in a PDS. The complete JCL file is to be used as SYSIN in the PDS update utility: IEBUPDTE.

Note: Because of this default option, all ’./’ characters found in JCL modules containing IEBUPDTE were replaced with ’:/’.

Once the JCL is split, the replacement must be performed the other way round before executing jobs which contain IEBUPDTE.

REPORT

BVPMMJCL produces a list for each JCL module created, with the parameters taken into account and according to the variants requested.

Note:

Since the parameters of the skeleton JCL are formatted as $xxxx, if BVPMMJCL encounters, upon execution, a $ character which does not correspond to a defined parameter, it sends error messages such as: ’Unknown symbolic parameter’ or ’Invalid position or length’ or ’Syntax error in symbolic parameter’.

These messages do not stop the execution and should be ignored: they apply to ’$’ characters present in the flow processed by BVPMMJCL but which are NOT parameters.

List of JCLs

Table of installation JCLs

Member	Contents	Procedure
D01ALLOM	Allocation of parameters PDSs	
D02CPARM	Loading of parameters	

Member	Contents	Procedure
D05IPROM	Proclib allocation (optional)	
D05PROCM	Loading of batch procedures in Proclib	
D06SKELM	Loading of SC skeleton file	
D07AEM	Loading of error messages	INAM
D99INSM	List of components' dates	INSM

JCL parameters

Syntax

===PRM PPPP=pppp .Comments

- Parameter values which contain special characters must be entered between simple quotes.
- Comments on ===PRM lines must not exceed column 72

  They must be preceded by a dot ('.').

Note: When the default value or the '<>' value is indicated, the parameter is required.

Important advice:

You are strongly advised to install the product in an environment totally distinct from the VA Pac environment.

This package contains programs and files already delivered with VA Pac, and compatibility problems may arise from an installation in a common environment.

Specific parameters have been added in order to take this constraint into account. These are the INDMV, INDMN, VOLMN, VOLMV and UNITMN parameters described in the following table.

Table of parameters

Parameter	Meaning	Default
	JOB lines:	
PRFJ	Jobname prefix (maximum: 5 characters)	PAC
CCPT	Job accounting code	<>
CLASSJ	Job execution class	1
MSGCL	JCL output class	A

Parameter	Meaning	Default
BASE	Development Database code	BASE
OUT	SYSOUT print class	H
INDSV	VA Pac VSAM system (maximum: 24 characters)	'EXP.BVPPV'
INDUV	VSAM user (maximum: 24 characters)	'UTI.BVPPV'
INDMV	VSAM migration (maximum: 24 characters)	'MIG.BVPMN'
INDMN	Non-VSAM migration (SAM, PDS) (maximum: 24 characters)	'MIG.BVPMN'
VOLUN	VOL=SER for NON-VSAM user files	
VOLMN	VOL=SER for NON-VSAM migration files	
VOLMV	VOL=SER for VSAM migration files (2)	
VCAT	VSAM catalog of the Development Database (User files)	
SCAT	System VSAM catalog (System files)	
LSK	Skeleton language (1)	A
UWK	UNIT of work files used	SYSDA
UNITUN	UNIT of NON-VSAM system files	SYSDA
UNITMN	UNIT of NON-VSAM migration files	SYSDA
HLQ	Common prefix of elements delivered with SMP/E (maximum 30 characters)	'HLQ'
BIBP	DSNAME of the procedure library	'SYS1.PROCLIB'
BIBT	DSNAME of sort library (SORTLIB)	'SYS1.SORTLIB'
BCOB	DSNAME of COBOL routine library	'SYS1.SCEERUN'
	Update of CICS CSD	

(1) A = English, F = French

(2) Required parameter, even under SMS.

Separators of JCL modules

Lines before and after JCL modules

```
===BEGMOD
....1    )
.....    ) Lines to be inserted before each JCL module
....n    )
===ENDMOD
....1    )
```

```
.....    ) Lines to be inserted after each JCL module
....n    )
```

Lines may be inserted as input to BVPMMJCL if the default option is not appropriate (see the installation default values presented above).

The purpose of these lines is to split the JCL file created by the BVPMMLCL utility into as many members as JCL modules.

This utility adds ....1 to ....n lines in front of each JCL module and ....1 to ....n lines to the end of each JCL module.

Installation of components

Allocation and loading of parameters

D01ALLOM module: '$prfj.D1' job

Allocation of file: $INDMN..BVPSY

Step	Program	Comments
STEP1	IDCAMS	DELETE of files
STEP2	IEFBR14	allocation of files

Execution JCL

```
//$PRFJ.D1   JOB ($CCPT),'PAC D01ALLOM',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D01ALLOM                             *
//* INITIAL ALLOCATION OF THE PARAMETER PDS AND ADDITIONAL FILES      *
//*     .STEP1 : LISTCAT                                              *
//*     .STEP2 : ALLOCATION                                           *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  LISTC ENT($INDMN..BVPSY)
/*
//STEP2  EXEC PGM=IEFBR14,COND=(0,EQ,STEP1)
//SY       DD DSN=$INDMN..BVPSY,
//         DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLMN,
//         UNIT=$UNITMN,
//         SPACE=(6080,(50,,5))
//
```

D02CPARM module: '$prfj.D2D' job

Loading of $INDMN..BVPSY file

Step	Program	Comment
STEP1	IEBUPDTE	Loading of the PDS members

Caution:

Replace :/ with ./ before submitting the JOB.

Execution JCL

```
===FRM TYPE=DATA
//$PRFJ.D2D  JOB ($CCPT),'PAC D02DPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D02DPAR                              *
//*    LOADS PDS OF COMMON PARAMETERS                                 *
//*     .STEP1 : LOADING COMMON PARAMETERS                            *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDMN..BVPSY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/     ADD NAME=DFBVPAE
 DELETE ($INDMV..BVPAE) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDMV..BVPAE)           -
                  SHR (2,3) RUS   KEYS (12,0)    -
                  VOL ($VOLMV)                   -
                  CYL (45)                       -
                  RECSZ (80 80) )                -
        INDEX   ( NAME ($INDMV..BVPAE.I)         -
                  CISZ (4096) )                  -
        DATA    ( NAME ($INDMV..BVPAE.D)         -
                  FSPC (2,1)                     -
        CISZ (4096) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSC
 DELETE ($INDMV..BVPSC) CLUSTER
   DEFINE CLUSTER ( NAME ($INDMV..BVPSC)         -
                  SHR (2,3) RUS   KEY (4 0)      -
                  VOL ($VOLMV)                   -
                  RECSZ (3204,3204) )            -
        INDEX   ( NAME ($INDMV..BVPSC.I)         -
                  CISZ (4096) )                  -
```

```
        DATA    ( NAME ($INDMV..BVPSC.D)          -
                  FSPC (10,5) SPEED  TRK (5)      -
        CISZ (3584) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFSYSEXT
 DELETE (&USES.SYSEXT.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME (&USES.SYSEXT.&USER)     -
                  SHR (2 3)  RUS  KEYS(43 0)      -
                  VOL ($VOLMV)                    -
                  CYL (5 1)                       -
                  RECSZ (254 254) )               -
        INDEX   ( NAME (&USES.SYSEXT.&USER.I)   -
                  CISZ (4096) )                   -
        DATA    ( NAME (&USES.SYSEXT.&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DLSYSEXT
 DELETE (&USES.SYSEXT.&USER) CLUSTER
:/     ADD NAME=MAXKEY
99999999999999999999999999999999999999999999999999999999999999999999
:/     ADD NAME=KEY01
00000001
:/     ADD NAME=REPRO999
 REPRO INFILE (MAXKEY)   OUTFILE (SYSPAF)
:/     ADD NAME=REPRO
 REPRO INFILE (IN1)    OUTFILE (OU1)
:/     ADD NAME=VERIFAE
 VERIFY FILE (PAC7AE)
:/     ADD NAME=VERIFSC
 VERIFY FILE (PAC7SC)
PP
//
```

Loading of procedures

It is recommended that all operation procedures be cataloged in one procedure library:

- Either in a reserved PROCLIB: in this case, execute the allocation job first, and then the loading job.
- Or in an existing PROCLIB: in this case, execute the loading job straight away.

D05IPROM module: '$prfj.D5I' job

Allocation of a reserved library (optional)

Step	Program	Comments
STEP1	IEFBR14	Allocation of a procedure library

Execution JCL

```
//$PRFJ.D5I  JOB ($CCPT),'PAC D05IPROC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D05IPROM                             *
//*                                                                   *
//*                WARNING! OPTIONAL JOB                              *
//*                ====================                               *
//*                                                                   *
//*     INITIAL ALLOCATION OF A SPECIAL "PROCLIB" FOR THE PRODUCT     *
//*     .STEP1 : LISTCAT                                              *
//*     .STEP2 : ALLOCATION                                           *
//*                                                                   *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  LISTC ENT($BIBP)
/*
//STEP2  EXEC PGM=IEFBR14,COND=(0,EQ,STEP1)
//LIB      DD DSN=$BIBP,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLMN,
//         UNIT=$UNITMN,
//         SPACE=(6080,(200,20,10))
```

D05PROCM module: '$prfj.D5M' job

Loading of procedures

This job includes an IEBUPDTE step, which creates one member for each procedure.

Caution:

Replace all :/ with ./ before submitting the job.

Each member is coded 'BVPNNNN', where NNNN is the standard name of the procedure.

Execution JCL

```
//$PRFJ.D5M JOB ($CCPT),'PAC D05PROC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D05PROCM                             *
//*                                                                   *
//*    CATALOGING OF BATCH PROCEDURES CONVERGENCE                     *
//*                                                                   *
```

```
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*                                                                   *
//*******************************************************************
//UPD    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$BIBP,DISP=SHR
//SYSIN    DD DATA,DLM='F+'
:/ ADD NAME=BVPINAM
:/ ADD NAME=BVPINSM
:/ ADD NAME=BVPMIBA
:/ ADD NAME=BVPMIMA
:/ ADD NAME=BVPPACX
:/ ADD NAME=BVPUTCG
:/ ADD NAME=BVPUTCR
:/ ADD NAME=BVPUTMA
:/ ADD NAME=BVPUTOV
:/ ADD NAME=BVPUTSE
F+
//
```

Loading of generation skeletons

D06SKELM module: '$prfj.D6' job

Creation and loading of skeleton files.

Step	Program	Comments
STEP1	IDCAMS	SC DELETE/DEFINE
STEP2	IDCAMS	SC loading (REPRO)

Execution JCL

```
//$PRFJ.D6  JOB ($CCPT),'PAC D06SKELM',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*******************************************************************
//* VISUALAGE PACBASE                                               *
//*                                                                 *
//*               INSTALLATION - D06SKELM - CONVERGENCE             *
//*   LOADING OF PRODUCT SKELETON FILES FOR GENERATION              *
//*    .STEP1 : DELETE DEFINE SKELETON FILES SC                     *
//*    .STEP2 : LOADING VSAM SKELETON FILES SC                      *
//*******************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDMN..BVPSY(DFBVPSC),DISP=SHR
//*
//STEP2  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SCO      DD DSN=$INDMV..BVPSC,DISP=SHR
```

```
//SCI      DD DSN=$HLQ..SBVPMF4(BVPSC$LSK),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (SCI)   OUTFILE (SCO)
//
```

Loading of error messages and online help

D07AE0 module: '$prfj.D7' job

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

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

To take one or more languages into account, you must add as many parameter lines which contain the two-characters language code in position 3.

To take all the available languages into account, you must enter a parameter line with '**' in position 3.

The AE file must be closed in on-line mode.

Step	Program	Comments
INPUT	BVPTU001	Recognition of the language parameter
DELDEF	IDCAMS	DELETE DEFINE of the AE file
PTUCAE	BVPTUCAE	Loading of the AE file supplied

Execution JCL

```
//$PRFJ.D7  JOB ($CCPT),'PAC D07AEM',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D07AEM                               *
//*********************************************************************
//INAM   EXEC BVPINAM
  **
/*
```

List of components' dates

D99INSM module: '$prfj.D99' job

This job produces the following lists:

- The list of batch programs with their generation characteristics,
- The list of AE and SC files with their creation dates.

In case of system operation problems, these lists should be used in order to communicate all the installation references to the Support.

This job also executes the INSM procedure.

Meaning of the return code on a component:

Return code	label
6	Load-module not found in the library
A	Not standard

Execution JCL

```
//$PRFJ.D99  JOB ($CCPT),'PAC D99INSM',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D99INSL                              *
//*    THE FOLLOWING JOB PROVIDES THE LIST OF PROGRAMS                *
//*    AND SYSTEM FILES INSTALLED ON THE SITE.                        *
//*********************************************************************
//INSL   EXEC BVPINSM
/*
```

Chapter 2. Migration procedures

UTCG - Repository analysis

UTCG - Overview

Principle

This procedure is a utility program that is used to list entities that can be generated in a current session, with all libraries taken into account. That is to say, Programs and Macros, Screens and Dialogues (from all types), eBusiness entities, Database Blocks.

Execution conditions

None

Result

Each generable entity found is loaded in the PAC7RQ sequential file.

Structure of the PAC7RQ file

Pos.	Len.	Value	Meaning
1	3		Library code
4	3	pgm, mac, dia, scr, ebu, dbd lib, met	Entity type
7	3		Type of Macro or type of eBusiness entity or type of Screen
		pgm, dia, ebu, mix	Macro
		ce, cs, do, mc, si	eBusiness entities
		fic, obj, rel	Meta-entities
10	30		Entity code
40	2		Variant or type of Block
42	1		Usage in the -CE desc. of a Screen
		'1'	Screen entity (scr) used in the -CE of another Screen
43	1		Option for inserting comments on a Library definition

Pos.	Len.	Value	Meaning
44	2		Generation variant for an eBusiness entity
46	1		Presence indicator of EXEC PAF order

UTCG - Description of steps

Repository analysis : BVPLTCGE

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development database data
PAC7AN	&INDUV..&BASE.AN	Input	Development database index
PAC7RQ	&UTCGFILE	Output	List of generable entities

UTCG - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*                 CARTOGRAPHIE GENERALE
//*
//* ------------------------------------------------------
//*
//BVPUTCG  PROC BASE=$BASE,               CODE OF DEVPT DATABASE
//         INDUV='$INDUV',              INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',                   USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                 SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUT=$OUT,                                OUTPUT CLASS
//         UTCGFILE=,               OUTPUT FILE FOR CARTOGRAPHIE
//         VOLS='SER=$VOLUN',                     BACKUP VOLUME
//         UNITS=$UNITUN,                           BACKUP UNIT
//         SPAOT='(TRK,(500,80),RLSE)'          OUTPUT FILE SPACE
//*************************************************************
//PLTCGE EXEC PGM=BVPLTCGE
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7RQ   DD DSN=&UTCGFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=15000,LRECL=150)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

UTSE - Analysis of Data Structures/Segments

UTSE - Overview

Principle

This procedure is a utility program that is used to analyze the Data Structures and Segments available in all the Libraries of the database.

Analysis of Data Structures:

Search for type

Search for their usages:

- Calls from -CD descriptions

  Organization, access mode, recording mode, opening, usage, break level, synchronization level, description type, cobol level, selection of Segments of type *01=02
- Calls from a working

  Mode of description, organization
- Calls from the Structured language of EBusiness entities

Analysis of Segments:

Search for particular usages such as redefines and Segments calls.

Search for usages in Database blocks

Search for -GG description (SQL complement)

Search for presence of sub-schema/sub-system

Search for usages in Screens (-CS).

Search for usages in Ebusiness entities.

Execution conditions

None

Result

Each Data sructure and each Segment is loaded in the PAC7RQ sequential file.

Structure of the PAC7RQ file

Pos.	Len.	Value	Meaning
1	3		Library code
4	3	DST or SEG	Entity type
7	4		Code of the DS or Segment
11	25		DS characteristics
11	1		Type of Data Structure
		G, T, M, N	Pactables
		V	Logical view
		Z	Default type
12	1		Organization
13	1		Access mode
14	1		Record mode
15	1		Opening
16	1		Number of break level
17	1		Number of synchronization levels
18	1		File usage
19	1		Type of generated description
20	1		level
21	6		Code of the calling program
27	9		Usage in a working storage section
27	1		Type of generated description
28	1		Level
29	1		Organization
30	6		Code of the calling program
36	1		Segment selection in the argument
		'1'	Selection in the *01=02 form
		blank	other
37	7		Segment characteristics
37	1		Used in a redefines
		'1'	Presence of a redefines
		blank	No redefines
38	1		Segment call

Pos.	Len.	Value	Meaning
		'1'	Segment called
		blank	Segment not called
39	1		Presence of -GG (SQL complement)
		'1'	Presence
		blank	absence
40	1		Used in a hierarchical Database block
		'1'	Used
		blank	Not used
41	1		Used in a Codasyl Database block
		'1'	Used
		blank	Not used
42	1		Used in a relational Database block (DB2)
		'1'	Used
		blank	Not used
43	1		Presence of sub-schema/ sub-system
		'1'	Presence
		blank	Absence
44			Call of Segment in a Screen (-CS)
44	6		Code of the Screen
50	1		Limitation of generation
51	1		Use of the file in reception
52	1		Use of the file on display
53	1		Category
54	4		Code of previous Segment
58	14		Access key input field
72	6		Data Elem. code for Segment access
78	1		Organization
79	1		Type of description
80	8		External name of the file
88	4		Code of Segment in the database
92	1		Break on display
93	2		Level of generated processes
95	1		Number of sub-schema

Pos.	Len.	Value	Meaning
96 96			Call of the DS or of the Segment in an EBusiness entity
96	2		Call type of the UE
98	6		Internal code of the UE
104	2		Type of description of the UE
106	6		Internal identifier

UTSE - Description of steps

Analysis of Data Structures, Segments: BVPLTSSE

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Data of the development database
PAC7AN	&INDUV..&BASE.AN	Input	Index of the development database
PAC7RQ	&UTSEFILE	Output	List of Data Structures and Segments

UTSE - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*                 ANALYSE DES SD ET SEGMENTS
//*
//* -------------------------------------------------------
//*
//BVPUTSE  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUT=$OUT,                                OUTPUT CLASS
//         UTSEFILE=,                OUTPUT FILE FOR CARTOGRAPHIE
//         VOLS='SER=$VOLUN',                     BACKUP VOLUME
//         UNITS=$UNITUN,                           BACKUP UNIT
//         SPAOT='(TRK,(500,80),RLSE)'          OUTPUT FILE SPACE
//*************************************************************
//PLTSSE EXEC PGM=BVPLTSSE
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
```

```
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7RQ   DD DSN=&UTSEFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=15000,LRECL=150)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

UTMA - Analysis of Macro-structures

UTMA - Overview

Principle

This procedure is a utility program that is used to analyze Macro-structures so as to identify those that are likely to bring about problems.

A macro-structure is a P-type entity (Program) that is used in Programs or Screens (-XP, -XO).

Execution conditions

None

Result

An output file PAC7RQ.

Structure of the PAC7RQ file:

Pos.	Len.	Value	Meaning
1	3		Library code
4	3	'MAC'	
7	3	'PGM' or 'SCR' or 'DEB'	-XP or -XO or -B
10	6		Macro code
16	2		Function code
18	2		Sub-function code
20	3		Line number
23	3		Operator with a parameter
26	3		Value of the operator in the PGM and SCR calls (col.3) with an occurs clause set to 10

Pos.	Len.	Value	Meaning
56	3		Operator (GT OR GB) or condition type (AN or OR) whose level or starting condition are not included in the Macro
59	32		= ORPHAN CONDITION if (AN or OR operator otherwise level of GT or GB)
91	1		Entity type for macros whose type is different from 'N' or 'M'
92	1		= 'N' when the function or sub- functions include a title line

Cases to be detected

- in some macros, a parameter can be indicated such as the function level, but in such cases, the -P lines are not detected.
- $ parameter for operators with the allowed values for $ parameter
- $ parameter declared in the section and/or paragraph of a -B description,
- lines -P with an orphan OR or AN type operator,
- macros whose entity type is different from 'N' or 'M',
- functions or sub-functions including several title lines ('N'),
- Detection of possible problematic operators: OPE, R, Dxx, DBL, SCn, SRO, SWH, SQL, MV + D.B.M.S operators.

UTMA - Description of steps

Analysis of Macro-structures: BVPLTMAC

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Data of the development database
PAC7AN	&INDUV..&BASE.AN	Input	Index of the development database
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	
PAC7RQ	&UTMAFILE	Output	List of generable entities

UTMA - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*                  MACRO ANALYSIS
//*
//* -------------------------------------------------------
```

```
//*
//BVPUTMA  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',                         SORT LIBRARY
//         LSR='BLSR',                     LSR BATCH SYSTEM NAME
//         OUT=$OUT,                                OUTPUT CLASS
//         UTMAFILE=,                OUTPUT FILE FOR CARTOGRAPHIE
//         VOLS='SER=$VOLUN',                     BACKUP VOLUME
//         UNITS=$UNITUN,                           BACKUP UNIT
//         UWK=$UWK,                              WORK FILES UNIT
//         SPAOT='(TRK,(500,80),RLSE)'          OUTPUT FILE SPACE
//*************************************************************
//PLTMAC EXEC PGM=BVPLTMAC
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7RQ   DD DSN=&UTMAFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=15000,LRECL=150)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

UTCR - Analysis of Data Elements

UTCR - Overview

Principle

This procedure is a utility program that is used to list Data Elements defined or undefined in the Repository from all Libraries. This utility makes an inventory of all information on these Data Elements (lists of values, child Data Element) as well as all usage formats.

Execution condition

None

Result

Each Data Element found is loaded in the sequential file PAC7RQ.

Structure of the PAC7RQ file

Pos.	Len.	Value	Meaning
1	3		Library code
4	3	'DEL'	Entity type
7	6		Data element code
13	1		presence indicator in the Reposit.
		'D'	Defined
		'N'	Undefined
14	1		Data element type
15	1		Data element internal usage
16	10		Internal format
26	10		Input format
36	14		Output format
50	1		Presence indicator of a list of values, applies to lines of D, P, O, I and SPACE types
		'1'	Presence
		blank	Absence
51	1		Presence indicator of addit.formats, applies to lines of F or E type
		'1'	Presence
		blank	Absence
52	1		Presence indicator of specif. labels, applies to lines of L, C or R types
		'1'	Presence
		blank	Absence
53	1		Presence indicator of specific data (of Alias and/or Reverse type), applies to lines of S, A or 8 types
		'1'	Presence
		blank	Absence
54	1		Presence indicator of eBusiness data, applies to lines of Y or G type
		'1'	Presence
		blank	Absence
55	1		Usage indicator for undefined Data elements

Pos.	Len.	Value	Meaning
		'R'	in a Report
		'S'	in a Segment
		'B'	in a Block
		'P'	in the working of a Program
		'O'	in a the working of a Screen
56	6		Code of the Data element that uses the current Data element
62	48		Content of the working declaration
110	6		Code of the parent Data element

UTCR - Description of steps

Analysis of Data elements: BVPLTRUB

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Data of the developement database
PAC7AN	&INDUV..&BASE.AN	Input	Index of the development database
PAC7RQ	&UTCRFILE	Output	List of Data elements

UTCR - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*                 ANALYSE DES RUBRIQUES
//*
//* ------------------------------------------------------
//*
//BVPUTCR  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUT=$OUT,                                OUTPUT CLASS
//         UTCRFILE=,               OUTPUT FILE FOR CARTOGRAPHIE
//         VOLS='SER=$VOLUN',                     BACKUP VOLUME
//         UNITS=$UNITUN,                           BACKUP UNIT
//         SPAOT='(TRK,(500,80),RLSE)'          OUTPUT FILE SPACE
//*************************************************************
//PLTRUB EXEC PGM=BVPLTRUB
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
```

```
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7RQ   DD DSN=&UTCRFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=15000,LRECL=150)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

UTOV - Analysis of Overrides

UTOV - Overview

Principle

This procedure is a utility program that is used to detect overrides of entity definitions and of their complete folders within the same library network. That is to say, on one hand you may find multiple occurences of the definition and of its folder, or on the other hand, one or more parts of its folder in a library different from that of the definition. The overrides analysis is possible only in the current view of the database.

Execution condition

None

Result

Each time an override is detected, the current entity and library are loaded in the sequential file PAC7OV. When the analysis of an entity family is ended, the total number of entities is saved in the PAC7OV file.

Structure of the PAC7OV file

Pos.	Len.	Value	Meaning
1	3	del, seg, dst, pgm, pia	Type of entity
4	30		Entity code
34	24		Folder record identifier (part of the index B3B9), blank for definitions
58	2		Data card code

Pos.	Len.	Value	Meaning
60	3		Reference library
63	3		Override libraries
66	1		Type of override
		'0'	For the definition
		'1'	For the folder itself
67	8		Total number of entities by type; when the counter is running, the type is filled-in, the other fields are blank

Pinted output

The total number of detected overrides is printed.

UTOV - Description of steps

Detection of overrides: BVPTUOVR

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error labels
PAC7AR	&INDUV..&BASE.AR	Input	Data of the development database
PAC7AN	&INDUV..&BASE.AN	Input	Index of the development database
PAC7OV		Report	Report
PAC7OV	&UTCGFILE	Output	List of detected overrides

UTOV - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*                 OVERRIDES DETECTION
//*
//* -------------------------------------------------------
//*
//BVPUTOV  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUV='$INDUV',              INDEX OF USER VSAM FILES
//         INDMV='$INDMV',            INDEX OF VSAM AE SC FOR MIG
//*:       VSAMCAT='$VCAT',                   USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                 SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUT=$OUT,                                OUTPUT CLASS
//         UTOVFILE=,               OUTPUT FILE FOR CARTOGRAPHIE
//         VOLS='SER=$VOLUN',                    BACKUP VOLUME
//         UNITS=$UNITUN,                          BACKUP UNIT
```

```
//         SPAOT='(TRK,(500,80),RLSE)'          OUTPUT FILE SPACE
//*************************************************************
//PTUOVR EXEC PGM=BVPTUOVR
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//PAC7AE   DD DSN=&INDMV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7ET   DD SYSOUT=&OUT
//PAC7OV   DD DSN=&UTOVFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=740,LRECL=74)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 3. Retrieval Procedures

MIBA - Batch Migration

MIBA - Overview

Principle

This procedure is a utility which prepares the batch migration after the extraction of entities or library contents in the Repository. It formats a file which will be used by the migration tools.

MIBA - User input

The user input of the MIBA 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	bbb	Extraction Library code
22	4	nnnn	Session number (Blank=current session)
26	1	'T'	Session status if Test session
29	4	EXLI ou EXTR (1)	Extractor code
34	1	'1'	Formatting for UPDP (PAF)
55	1	'1'	Timestamp request indicator

(1) If EXTR, 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-level libraries with the generation of identification lines ('*' lines)

Pos.	Len.	Value	Meaning
7	33	Choice	Entity to be extracted, coded in the same way as the 'Choice' field in TP.
40	4		Extraction type:
		'ALL '	Entity and used entities

For specific selections, refer to the documentation of the PACX procedure in the 'Adminstrator's Procedures' Manual.

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 Macro-Structures but not defined as such in the Repository.

This file will be completed subsequently to the exploitation of results of the UTCG Repository Analysis procedure.

This procedure also allows to replace unknown special characters in an entity code with another character. The correspondence table is represented by the PAC7CS file used as input to the BVPACS96 program (optional file, supplied by the client).

This file contains a 20-character-long record, constituted of 10 items, each of which is 2 characters long: the first represents the character to be replaced, the second is the replacing character.

MIBA - Description of steps

Input recognition: PTU001

Initialization of the KSDS work file: IDCAMS

Extraction: PACX

This step extracts transactions according to user input.

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7AN	&INDUV..&BASE.AN	Input	Development Database Index file
PAC7AR	&INDUV..&BASE.AR	Input	Development Database Data file

Code	Physical name	Type	Label
PAC7AY	&INDUV..&BASE.AY	Input	Development Database Extension Data
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGU	&INDSV..BVPGU	Input	Administration Database Users
PAC7PJ	DUMMY	Input	Archived transactions
PAC7MB	&&PACXMB	Input	User input
PAC7MA	DUMMY	Input	CPSN Master file
PAC7ES	DUMMY	Input	CPSN Slave file
PAC7BM	&&PACXBM	Input/Output	User input
PAC7MM	&&PACXMM	Input/Output	EXPU Work file
PAC7MJ	&&PACXMJ	Input/Output	EXPJ Work file
PAC7TE	&&PACXTE	Input/Output	RMEN Work file
PAC7RE	&&PACXRE	Input/Output	RMEN Work file
PAC7RM	&&PACXRM	Input/Output	RMEN Work file
PAC7WD	&&PACXWD	Input/Output	Extracted transactions
SYSEXT	&INDUV..SYSEXT.&USER	Input/Output	Work file
PAC7MV	&&MV	Output	Extracted transactions for UPDT
PAC7MR	&&MR	Output	Extracted transactions for REOR (EXPU)
PAC7MX	&&MX	Output	Non extracted entities (PACX)
PAC7GY	&&GY	Output	Extracted transactions for UPDP
PAC7TD	&&TD	Output	Extracted transactions for CPSN
PAC7UE	&&UE	Output	Extracted transactions for EXUE
PAC7IA		Report	General printout of the program stream
PAC7DD		Report	Errors on input transactions
PAC7ED		Report	Extractions report
PAC7EE		Report	Extractions report

Code	Physical name	Type	Label
PAC7EG		Report	Extractions report
PAC7EM		Report	Extractions report
PAC7EP		Report	Extractions report
PAC7EQ		Report	Extractions report
PAC7EU		Report	Extractions report
PAC7EZ		Report	Extractions report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in EXTR/EXUE extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD) or in EXLI (Database not available)

Deletion of the KSDS work file: IDCAMS

Batch migration : BVPACS96

Code	Physical name	Type	Label
PAC7GY	&&GY	Input	File resulting from PACX extraction
PAC7PM	&INPUTMIB	Input	File containing the list of the Pgms to be transformed into Macros (length = 50)
PAC7CS	&INPUTSPE	Input	Special characters correspondence file (length = 20)
PAC7TA	&&PAC7TA	Output	

Reorganisation of transactions : BVPACS97

Code	Physical name	Type	Label
PAC7TA	&&PAC7TA	Input	File containing transactions grouped by type
PAC7AT	&MIBAFILE	Output	

MIBA - Execution JCL

```
//* -----------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -----------------------------------------------------
//*                 MIGRATION BATCH
//*
//* -----------------------------------------------------
//*
//BVPMIBA  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDSV='$INDSV',              INDEX OF SYSTEM VSAM FILES
//         INDUV='$INDUV',                INDEX OF USER VSAM FILES
//         INDMN='$INDMN',   INDEX OF NON VSAM FILESFOR MIGRATION
//*:       VSAMCAT='$VCAT',                     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                   SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         STEPLIP=STEPLIP,     LIBRARY OF LOAD-MODULES VAPACBASE
//         SORTLIB='$BIBT',                           SORT LIBRARY
//         USER=,                                       USER CODE
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUT=$OUT,            UTILITIES AND ERRORS OUTPUT CLASS
//         OUTL=$OUT,                      OUTPUT CLASS OF REPORTS
//         UWK=$UWK,                                    WORK UNIT
//         SPAMB='(TRK,(5,1),RLSE)',           REQUEST FILE SPACE
//         SPAMV='(TRK,(50,10),RLSE)',            UPDT FILE SPACE
//         SPAGY='(TRK,(50,10),RLSE)',            UPDP FILE SPACE
//         SPATD='(TRK,(50,10),RLSE)',            CPSN FILE SPACE
//         INPUTMIB=,                          PROGRAM/MACRO FILE
//         INPUTSPE=,                           SPECIAL CHAR FILE
//         MIBAFILE=,                        OUTPUT FILE MIGRATION
//         VOLS='SER=$VOLUN',                       BACKUP VOLUME
//         UNITS=$UNITUN,                             BACKUP UNIT
//         SPAOT='(TRK,(500,80),RLSE)'          OUTPUT FILE SPACE
//*************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MIBAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDMN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
```

```
//SYSIN    DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//MAXKEY  EXEC PGM=IDCAMS
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY   DD DSN=&INDMN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDMN..BVPSY(REPRO999),DISP=SHR
//*
//PACX   EXEC PGM=BVPACX,REGION=0K
//*-----------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=&STEPLIP,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7PJ   DD DUMMY
//PAC7IA   DD SYSOUT=&OUTL
//PAC7DD   DD SYSOUT=&OUTL
//PAC7ED   DD SYSOUT=&OUTL
//PAC7EE   DD SYSOUT=&OUTL
//PAC7EG   DD SYSOUT=&OUTL
//PAC7EM   DD SYSOUT=&OUTL
//PAC7EP   DD SYSOUT=&OUTL
//PAC7EQ   DD SYSOUT=&OUTL
//PAC7EU   DD SYSOUT=&OUTL
//PAC7EZ   DD SYSOUT=&OUTL
//PAC7MA   DD DUMMY
//PAC7ES   DD DUMMY
//SYLSR    DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//SYSEXT   DD SUBSYS=(&LSR,'DDNAME=SYLSR')
//PAC7MB   DD DSN=&&MIBAMB,DISP=(OLD,DELETE,DELETE)
//PAC7BM   DD DSN=&&PACXBM,DISP=(,DELETE),UNIT=&UWK,
//            DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7MM   DD DSN=&&PACXMM,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=113,BLKSIZE=11300)
//PAC7MJ   DD DSN=&&PACXMJ,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMV,
```

```
//            DCB=(RECFM=FB,LRECL=158,BLKSIZE=6320)
//PAC7TE   DD DSN=&&PACXTE,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=323,BLKSIZE=6460)
//PAC7RE   DD DSN=&&PACXRE,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=36,BLKSIZE=6012)
//PAC7RM   DD DSN=&&PACXRM,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7WD   DD DSN=&&PACXWD,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7MV   DD DSN=&&MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MR   DD DSN=&&MR,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MX   DD DSN=&&MX,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7TD   DD DSN=&&TD,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7GY   DD DSN=&&GY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAGY,
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)
//PAC7UE   DD DSN=&&UE,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=230,BLKSIZE=6440)
//SYSUDUMP DD SYSOUT=&OUT
//*
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDMN..BVPSY(DLSYSEXT),DISP=SHR
//PACROU DD DSN=&&DLSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*------------------------------
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DLSYSEXT,DISP=(OLD,DELETE)
//PACS96 EXEC PGM=BVPACS96
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
```

```
//*:        DD DSN=&VSAMCAT,DISP=SHR
//PAC7GY   DD DSN=&&GY,DISP=(OLD,PASS)
//PAC7CS   DD DSN=&INPUTSPE,DISP=SHR
//PAC7PM   DD DSN=&INPUTMIB,DISP=SHR
//PAC7TA   DD DSN=&&PAC7TA,DISP=(,PASS),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=27066,LRECL=347)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACS97 EXEC PGM=BVPACS97
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//PAC7TA   DD DSN=&&PAC7TA,DISP=(OLD,PASS)
//PAC7AT   DD DSN=&MIBAFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=27066,LRECL=347)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

MIMA - Macro Migration

MIMA - Overview

This procedure enables you to simulataneously generate and tag one or more MacroStructures at once.

You must indicate a '*' line (user code, password, library...) as input, as well as one or more command lines for each MacroStructure to be generated.

On this command, coded 'GCM', you can enter the following parameters:

- VARIAN=x, represents the 'type of code' of the Macro Definition. If this parameter is not specified, the Macro will be generated with the 'type of code' indicated on its Definition (Since this value is 'N' on most Macros, it is advised to always specify this parameter).
- OPTNUM=x, represents the 'COBOL numbering and alignment' option on the Macro Definition. By default, the Macro will be generated with the value specified on the Macro Definition. Warning: The value ' ' is significant (numbering requested).
- INSCOM=x, represents the 'Comments insertion option' on the Library Definition. By default, the Macro will be generated with the value specified on the Library Definition.

- LANGEN=x, represents the generated language of Programs on the Library Definition. By default, the Macro will be generated with the value specified on the Library Definition.

  Warning: All the Macro generation requests for a given '*' line must contain the same value for these last two parameters.

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

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
		VARIAN=x	To assign value 'x' to the 'type of cobol to generate'
		OPTNUM=x	To sassign value 'x' to the 'Cobol numbering option'
		INSCOM=x	To assign value 'x' to the 'Comments insertion option'
		LANGEN=x	To assign value 'x' to the Library generated language

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

MIMA - Description of steps

Input recognition: PTU001

Generation/Print commands : BVPACA10

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV..&BASE.AN	Input	Development Database index
PAC7AY	&INDUV..&BASE.AY	Input	Development Database extension
PAC7AJ	&INDUV..&BASE.AJ	Input	Development Database journal
PAC7AE	&INDMV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7ME	&&MIMAMB	Input	User transactions
PAC7MV	&&PAC7MV	Output	Update transactions
PAC7MG	&&PAC7MG	Output	Generation/Print commands

GCM specific preparation : BVPACA30

Code	Physical name	Type	Label
PAC7MG	&&PAC7MG	Input	Generation/Print commands
PAC7JG	&&PAC7JG	Output	Extractor command

Extractor for the generation of Programs: BVPACP30

Code	Physical name	Type	Label
PAC7AE	&INDMV..BVPAE	Input	Error messages
PACGGR	&INDSV..BVPGN	Input	Administration Database data
PACGGN	&INDSV..BVPGR	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV..&BASE.AN	Input	Development Database index
PAC7AY	&INDUV..&BASE.AY	Input	Development Database extension data
PAC7JG	&&PAC7JG	Input	User transactions
PAC7W1	&&PAC7W1	Output	Work file
PAC7W2	&&PAC7W2	Output	Work file
SORTWK01		Sort	

Code	Physical name	Type	Label
SORTWK02		Sort	
SORTWK03		Sort	

Preparation for batch generation : BVPACP40

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV..&BASE.AN	Input	Development Database index
PAC7W1	&&PAC7W1	Input	Work file
PAC7W2	&&PAC7W2	Input	Work file
PAC7W3	&&PAC7W3	Output	Work file
PAC7W4	&&PAC7W4	Output	Work file
PAC7KP	&&PAC7KP	Output	Work file
PAC7GI	DUMMY	Output	Work file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

COBOL generator : BVPACP80

Code	Physical name	Type	Label
PAC7SC	&INDMV..BVPSC	Input	Generation skeleton
PAC7W3	&&PAC7W3	Input	Work file
PAC7W1	&&PAC7W5	Output	Work file

Formatting of the generated flow : BVPACP82

Code	Physical name	Type	Label
PAC7AE	&INDMV..BVPAE	Input	Error messages
PACGGR	&INDSV..BVPGN	Input	Administration Database data
PACGGN	&INDSV..BVPGR	Input	Administraion Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7W1	&&PAC7W5	Input	Work file
PAC7W4	&&PAC7W4	Input	Work file
PAC7GP	&&PAC7GP	Output	Work file

Code	Physical name	Type	Label
PAC7EP	&&PAC7EP	Output	Work file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Sort before tag insertion : BVPACP84

Code	Physical name	Type	Label
PAC7W1	&&PAC7W5	Input	Work file
PAC7W2	&&PAC7W6	Output	Work file

Sort and preparation before Macro tagging: BVPACP86

Code	Physical name	Type	Label
PAC7W1	&&PAC7W1	Input	Work file
PAC7RQ	&&PAC7RQ	Output	Work file

Generation of the tagget Macro : BVPACP87

Code	Physical name	Type	Label
PAC7GP	&&PAC7GP	Input	Work file
PAC7RQ	&&PAC7RQ	Input	Work file
PAC7W2	&&PAC7W6	Input	Work file
PAC7TA	&&MIMAFILE	Output	Work file

MIMA - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*                 MIGRATION MACROS
//*
//* -------------------------------------------------------
//*
//BVPMIMA PROC BASE=$BASE,                CODE OF VAPAC DATABASE
//         INDUV='$INDUV',              INDEX OF USER VSAM FILES
//         INDSV='$INDSV',            INDEX OF VSAM SYSTEM FILES
//         INDMV='$INDMV',           INDEX OF VSAM FOR MIGRATION
//*:       SYSTCAT='$SCAT',                 SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
```

```
//         STEPLIP=STEPLIP,     LIBRARY OF LOAD-MODULES VAPACBASE
//         MIMAFILE=,                        OUTPUT FILE MIGRATION
//         SPAOT='(TRK,(500,80),RLSE)',          OUTPUT FILE SPACE
//         VOLS='SER=$VOLUN',                        BACKUP VOLUME
//         UNITS=$UNITUN,                              BACKUP UNIT
//         SPAWK='(TRK,(50,10),RLSE)',             WORK FILE SPACE
//         SORTLIB='$BIBT',                           SORT LIBRARY
//         LSR='BLSR',                         LSR BATCH SYSTEM NAME
//         UWK=$UWK,                                     WORK UNIT
//         OUT=$OUT                                    OUTPUT CLASS
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MIMAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PACA10 EXEC PGM=BVPACA10
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=&STEPLIP,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//PAC7AJ   DD DUMMY
//PAC7AE   DD DSN=&INDMV..BVPAE,DISP=SHR
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7ME   DD DSN=&&MIMAMB,DISP=(OLD,PASS)
//PAC7MG   DD DSN=&&PAC7MG,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=15300)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=17000)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACA30 EXEC PGM=BVPACA30
```

```
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7AE   DD DSN=&INDMV..BVPAE,DISP=SHR
//PAC7MG   DD DSN=&&PAC7MG,DISP=(OLD,PASS)
//PAC7JG   DD DSN=&&PAC7JG,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//PAC7KU   DD DSN=&&PAC7KU,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACP30 EXEC PGM=BVPACP30
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//PAC7JG   DD DSN=&&PAC7JG,DISP=(OLD,PASS)
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//PAC7AE   DD DSN=&INDMV..BVPAE,DISP=SHR
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7W1   DD DSN=&&PAC7W1,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//PAC7W2   DD DSN=&&PAC7W2,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
```

```
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACP40 EXEC PGM=BVPACP40
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7W1   DD DSN=&&PAC7W1,DISP=(OLD,PASS)
//PAC7W2   DD DSN=&&PAC7W2,DISP=(OLD,PASS)
//PAC7W3   DD DSN=&&PAC7W3,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//PAC7W4   DD DSN=&&PAC7W4,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=90,BLKSIZE=18000)
//PAC7KP   DD DSN=&&PAC7KP,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//PAC7GI   DD DUMMY,DCB=BLKSIZE=80
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACP80 EXEC PGM=BVPACP80
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//PAC7SC   DD DSN=&INDMV..BVPSC,DISP=SHR
//PAC7W3   DD DSN=&&PAC7W3,DISP=(OLD,PASS)
//PAC7W1   DD DSN=&&PAC7W5,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//PACP82 EXEC PGM=BVPACP82
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//PAC7AE   DD DSN=&INDMV..BVPAE,DISP=SHR
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
```

```
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7W1   DD DSN=&&PAC7W5,DISP=(OLD,PASS)
//PAC7W4   DD DSN=&&PAC7W4,DISP=(OLD,PASS)
//PAC7GP   DD DSN=&&PAC7GP,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//PAC7EP   DD DSN=&&PAC7EP,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACP84 EXEC PGM=BVPACP84
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//PAC7W1   DD DSN=&&PAC7W5,DISP=(OLD,PASS)
//PAC7W2   DD DSN=&&PAC7W6,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=260,BLKSIZE=26000)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACP86 EXEC PGM=BVPACP86
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//SYSOUT   DD SYSOUT=H
//PAC7W1   DD DSN=&&PAC7W1,DISP=(OLD,PASS)
//PAC7RQ   DD DSN=&&PAC7RQ,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=50,BLKSIZE=1000)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACP87 EXEC PGM=BVPACP87
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//SYSOUT   DD SYSOUT=H
//PAC7GP   DD DSN=&&PAC7GP,DISP=(OLD,PASS)
//PAC7RQ   DD DSN=&&PAC7RQ,DISP=(OLD,PASS)
```

```
//PAC7W2   DD DSN=&&PAC7W6,DISP=(OLD,PASS)
//PAC7TA   DD DSN=&MIMAFILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAOT,
//         DCB=(RECFM=FB,BLKSIZE=27000,LRECL=1000)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACD90 EXEC PGM=BVPACD90,
//       PARM='  00000000000000000000000000000000'
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=&STEPLIP,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*        DD DSN=$DFHEXC,DISP=SHR
//PAC7AE   DD DSN=&INDMV..BVPAE,DISP=SHR
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7KU   DD DSN=&&PAC7KU,DISP=(OLD,PASS)
//PAC7KB   DD DUMMY
//PAC7KD   DD DUMMY
//PAC7KF   DD DUMMY
//PAC7KP   DD DUMMY
//PAC7ID   DD SYSOUT=&OUT
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Part Number: DEMCI000351A - 8129

Printed in USA