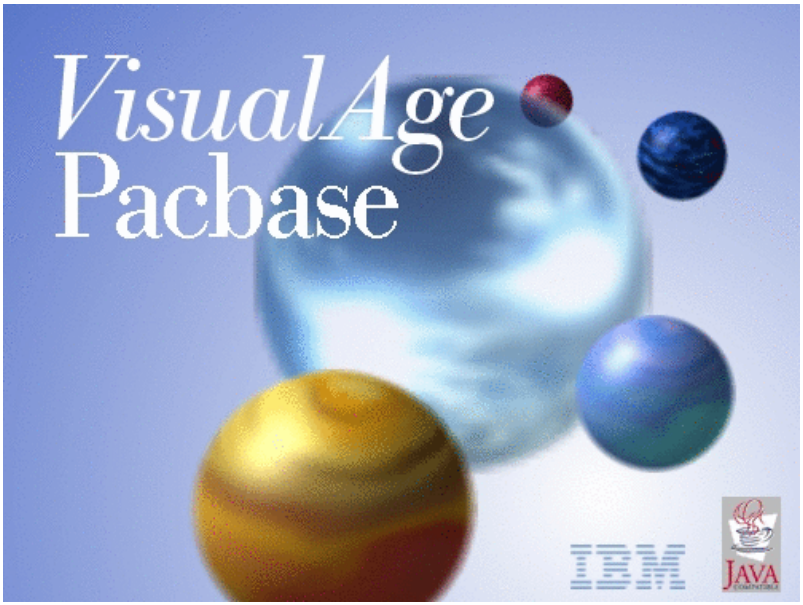
VisualAge Pacbase

Pactables - IBM MVS - IMS

Version 3.5



VisualAge Pacbase

Pactables - IBM MVS - IMS

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 (September 2006)

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.

Contents

Notices

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

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

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

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

Trademarks

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

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

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

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

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

Chapter 1. Foreword

How to use this manual for system installation

If a previous version of Pactables is already installed on the site, the 3.5 version must be different from any former Pactables version regarding the installation parameters. The test case provided on the installation media must be executed.

Chapter 2. Pactables components

Introduction

The purpose of the Pactables function is to process a certain amount of permanent data whether on-line or in batch mode (see the Pactables Reference Manual).

Two types of resources are therefore necessary:

- Libraries which store the programs making up the Pactables module, and its parameters.
- Permanent files, which contain the data processed by those programs. These files can be divided into two categories:
  - 'System' files, which remain stable during the use of the Pactables function,
  - 'Evolving' files, which are handled by the users, and whose volumes vary according to the types of updates performed.

Note:

Pactables is installed independently of the other VisualAge Pacbase functions.

The implementation of the Pactables function requires data which must be defined and described with the VisualAge Pacbase Specifications Dictionary function. The Extraction Procedure required to operate the Pactables function is described in the VisualAge Pacbase 3.5 Operations Manual.

The on-line programs library

On-line program library: SBVPMTR8

Code	Operation and meaning
BVPFT00	Tables extraction (TUF-TP MODULE)
BVPFT10	User parameters extraction (TUF-TP MODULE)
BVPFT90	User Interface (TUF-TP MODULE)
BVPP500	Enter, FT or 'clear' : initial screen
BVPP510	C1: read-write access to mono-item with CR, MO, DE for updating
BVPP512	C1: read-write access to mono-item with CR, MO, DE for updating

Code	Operation and meaning
BVPP520	C2: read-only access to multi-item with DE for an item deletion
BVPP522	C2: read-only access to multi-item with DE for an item deletion
BVPP530	LT : list of tables
BVPP540	LS : list of sub-schemas/sub-systems
BVPP550	LD : list of documentation
BVPP560	C3 : read-only access to item hist. acc.
BVPP570	HELP screen
BVPP580	LH : list of historical accounts
BVPP590	LJ, LE : print requests
BVPP599	Display of system errors
BVPP600	Parameter & password updating
BVPP610	User code updating
BVPP620	Access authorization updating
BVPP820	Optimized access module
BVPP920	Generalized access module
BVPPLNK	Pactables access via user-program module
BVPSECT	Security systems interface sub-program

Important

During updates, the BVPP510 and BVPP520 programs may call the user check routines in order to perform additional checks. As a default, the generation option of these routines is without the century management.

If the user check routines are generated with the century management option, the two programs (BVPP512 and BVPP522) must be renamed and used instead of BVPP510 and BVPP520.

In all cases, ALL user check routines should be generated with the same option.

The batch programs library

Batch program library : SBVPMBR8

Code	Proc.	Meaning
BVPTU001		Copy of transactions file
BVPTAINI	INTA	File initialization
BVPTARST	RSTA	Table restoration
BVPTARSV	-	- -
BVPTARSG	-	- -
BVPTASVT	SVTA	Table backup
BVPTASVV	-	- -
BVPTASVG	-	- -
-	RETA	Table reorganization
BVPTAU80	TUTA	Direct consultation of tables
BVPTA100	PMTA	Parameter update
BVPTA120	-	- -
BVPTA150	EXTA	Table extraction
BVPTA160	-	- -
BVPTA250	GETT	Table generation
BVPTA290	-	- /Lists -
-	LDTA	- /Lists -
BVPTA300	UPTA	Table update
BVPTA302	-	- -
BVPTA310	IMTA	Table import
BVPTA312	-	- -
BVPTA320	PRTA	Table printout
BVPTA350	UPTA	- -
-	IMTA	- -
-	PRTA	- -
BVPTA360	UPTA	- -
-	IMTA	- -
-	PRTA	- -
BVPTA400	RETA	Table reorganization
BVPTA410	-	- -
BVPTA420	-	- -

Code	Proc.	Meaning
BVPTA430	-	- -
BVPTAD05	CDT1	Table-description comparison
BVPTAD10	-	- - -
BVPTAD20	CDT2	- - -
BVPTAV10	CVTA	Table update
BVPTAV20	-	- -
BVPSECB		Security systems sub-program
BVPTATCD	TCTA	TD file sort
BVPTATCG	-	TG file sort
BVPTATCV	-	TV file sort
BVPTATC1	-	TC partitioning according to the type of file
BVPTABVS	BVTA	Loading of TD-TV databases onto VSAM files
BVPTAT10	LDTE	Loading of TE database - error messages
BVPTAT15	LDTG	Loading of TG database - user parameters
BVPTAT17	LDTZ	Loading of TZ database - Spas types
BVPTAT19	LDTB	Loading of TB database - TP TUF database
BVPTARE0	LDRE	Loading of TD TV after reorganization

Important note:

During updates, the BVPTA302 and BVPTA312 programs may call the user check routines in order to perform additional checks. The default generation option of these routines is 'without century management'.

If the user check routines are generated with the century-management option, the two new programs, BVPTA302 and BVPTA312, must be renamed and used respectively in the UPTA and IMTA procedures instead of the BVPTA300 and BVPTA310 programs.

In all cases, ALL the user check routines should be generated with the same century-management option.

Security systems interface extension

The optional BVPSECUR sub-program interfaces the Pactables function with the site's security system.

This sub-program is supplied by SMP/E in the hlq.SBVPMBR8 PDS.

Refer to the chapter 'Security system interface' in the VA Pacbase Installation Guide for details on this extension operations.

The VA Pac macro-structures library

TUF-TP macro-structures catalog

The Macro-structures are the following ones:

Code	Meaning
AATUFA	Description of the table data element
AATUFL	'LT' or 'LH' list
AATUFS	'LS' or 'LC' list
AATUFU	List of user parameters
AATUFX	List of table items

These macro-structures are to be used in user on-line application programs using the TUF-TP facility.

They allow to add the description of communication areas which are necessary to call the BVPFT90 sub-program of the TUF-TP facility.

These Macro-structures are supplied as VA Pac update transactions. They can be downloaded through the VA Pac Support web page at http://www.ibm.com/software/awdtools/vapacbase/support.html

They must be loaded into the VA Pac library used for the development of user transactions by taking the transactions of VA Pac UPDP or UPDT procedures as input.

The parameters library

THE PARAMETERS LIBRARY: SY

Its size is approximately 5 tracks (on a 3380 disk-unit).

It contains:

- The DEFINE and VERIFY of VSAM files:
  - The DELETE/DEFINEs of each Pactables VSAM file are named DF$BASE.xx (with xx=AD, AV, DD, TD, TG, TU, TV, TS, TW, T1 and TB), and DFBVPxx (with xx = TE, TZ),

    Example: DFT350TD or DFBVPTEE.
  - The VERIFYs of each Pactables VSAM files are named VERIFxx

- The initial record for the TG, TZ and TB databases:
  - LD$BASE.TG
  - LDBVPTZ
  - LD$BASE.TB
- The SYSIN used for TG loading:
  - SY$BASE.TG
- The BLDG of the TC generation file:

  The request to build the indexes of the Pactables TC backup file is listed under the name BL$BASE.TC.
- The DFSVSAMP:

  The VSAM Control Cards for buffers are listed under the name DFSVSAMn (n = 8 or 9). These cards are initialized with the standard values at installation, but their management is the responsibility of the Product Systems Manager.
- The 'APPLCTN' and 'TRANSACT' macro-instructions:

  The set of macro-instructions that were defined in the IMS Control Region are listed under PACTCTRL. This PDS is controlled by the system staff.

NOTE:

Any modification made on tables characteristics must be made in the SY library.

The system files

They represent the system itself. These files and databases are not modified by daily handlings, and they must be reloaded whenever the system needs to be re-installed.

- The library of batch load modules: SBVPMBR8.
- The library of on-line load modules: SBVPMTR8
- The parameters library (PDS): BVPSY.
- The database which contains the error messages and automatic HELP documentation of the Pactables function (TE):

Characteristic	Value
Size	Approx. 1,000 records
Organization	DL/1 HISAM
DSN	$INDSV..BVPTE
DBDName	PACDTE$SUG
Record length	98 bytes

Characteristic	Value
length	90 bytes
Use	Batch and on-line

Note: The error message database is loaded from a source file (IN):

Characteristic	Value
Size	Approx. 700 records
Organization	sequential
DSName	$INDSN..BVPIN
Record length	90 bytes
BLKSIZE length	1800 butes
Use	Batch

For the IMS version, the Pactables system uses two other libraries:

- a DBD library : SBVPDBD
- a PSB library : SBVPPSB

The evolving files

The evolving files contain the User information and are managed by the system, either in on-line or batch mode:

- Table descriptions database: TD

Characteristic	Value
Organization	DL/1 HISAM
DSName	&INDUV..$BASE.TD
DBDName	PACDTD$SUG
Segment length	240 bytes
Record length	248 bytes
Space	16 records per CI of 4,096
Use	Batch and on-line

- Table contents database: TV

Characteristic	Value
Organization	DL/1 HIDAM
DSName	&INDUV..$BASE.TV

Characteristic	Value
DBDName	PACDTV$SUG
Segment length	80 to 1100 bytes
Record length	4,089 bytes
Space	adapted to the length of the tables
Use	Batch and on-line

- TV primary indexes database: TU

Characteristic	Value
Organization	VSAM index database
DSName	&INDUV..$BASE.TU
DBDName	PACDTU$SUG
Segment length	35 bytes
Record length	40 bytes
Space	25 records per CI of 1,024
Use	Batch and on-line

- User parameters database: TG

Characteristic	Value
Organization	DL/1 HISAM
DSName	&INDUV..$BASE.TG
DBDName	PACDTG$SUG
Segment length	85 bytes
Record length	94 butes
Space	43 records per CI of 1,024
Use	Batch and on-line

TUF-TP work database: TB

Characteristic	Value
Organization	DL/1 HIDAM
DSName	&INDUV..$BASE.TB
DBDName	PACDTB$SUG
Segment length	80 to 1,140 bytes
Record length	4,089 bytes

Characteristic	Value
Space	adapted to the length of the tables
Use	TP

- TB primary indexed database: T1

Characteristic	Value
Organization	VSAM index database
DSName	&INDUV..$BASE.T1
DBDName	PACDT1$SUG
Segment length	63 bytes
Record length	68 bytes
Space	15 records per CI of 1,024
Use	on-line

  - Spas database: TZ

Characteristic	Value
Organization	VSAM index database
DSName	&INDUV..$BASE.T1
DBDName	PACDT1$SUG
Segment length	63 bytes
Record length	68 bytes
Space	15 records per CI of 1,024
Use	on-line

  - Backup file: TC

Characteristic	Value
Organization	Sequential with generation
DSName	&INDUN..$BASE.TC
Format	Variable
Length	1067 bytes
Use	Batch

This is a standardized Pactables backup file in a sequential format: descriptions (TD), contents (TV) and authorizations (TG).

Standard limitations

Maximum length for a table item : 999 characters

Maximum length for the table key : 20 characters

Maximum number of Data Elements in a table : 40

Number of table items per table : Unlimited

Chapter 3. Environment

Introduction

It is assumed that the site where Pactables is installed provides the environment and the resources required to run the system.

The purpose of this chapter is to define this environment, and thus help determine how much disk space is required.

File sizes are specified in the Chapter 'Prerequisites' of the VisualAge Pacbase 'Installation Guide : Server & Client components'

On-line environment

The Monitor used for TP Pactables is IMS/DC.

Since Pactables V2.5, the MFS system has been bypassed by the use of the DFS.EDTN standard module for the receiving and display of messages.

Messages are formatted by the BVPR980 sub-program located in the on-line programs library.

Two transaction codes are to be declared in IMS to access Pactables:

- $TRANT transaction code for a connection to Pactables,
- $TRANP transaction code for a connection to Adminstration Pactables.

The various Pactables on-line transactions to be declared in IMS are transactional and use a SPA with a length of 150.

Access methods

General principles

Pactables is its own DBMS. It uses DL/1 only to store the records that support its physical organization.

As a result:

- The user cannot directly access Pactables data via the DL/1 standard utilities. This can be done only via utilities provided with Pactables.
- DL/1 resources are seldom used by the Pactables function. In particular, all databases are made up of a single root segment.

There are no dependent segments and the DBRs have fixed length (except those belonging to the TV database and whose lengths are variable).

- Secondary indexes and logical relationships are never used. Therefore, the ratio 'physical accesses' / 'DL/1 accesses is considerably lower than in current applications.

Database organization

Two types of DL/1 Database organizations are used: HISAM and HIDAM.

HISAM-VSAM Databases

This organization is used for the following databases: table descriptions (TD), User Codes (TG), Error Messages and documentation (TE), and the SPAs Database (TZ).

The DL/1 physical record contain one and only one complete DBR, i.e. only one VisualAge Pacbase record plus DL/1 control data. There is no DATASET OVERFLOW. The size of the database is calculated directly from the number of logical records, the size of the record and the FREE SPACE requested during the DEFINE of the VSAM file.

HIDAM-VSAM Databases

This organization is used for the database that contains the table contents (TV) and whose primary index is built by the TU database. It is also used for the TUF-TP (TB) work database whose primary index is built by the database (T1).

Each DBR of these databases contains a single ROOT segment whose size may vary between 80 and 1,100 bytes for TV and between 80 and 1,140 bytes for TB.

Batch environment

In batch mode, the system uses standard functions of the operating system.

The memory size required for the execution of batch procedures vary mainly according to the size of the buffers allocated to the files used by these procedures.

Environment preparation

This step consists in preparing the environment for the installation of Pactables, i.e. :

- Choose the VSAM catalogs and provide for the necessary disk space,
- Prepare the generation of IMS/DC by taking the following parameters into account:

Parameters for the generation of IMS/DC

In the names herebelow, the symbols xx and rr represent a suffix and a root of your choice. However, the installation can be facilitated by using the following values for the DBD, PSB and Formats:

- xx = 35 for the suffix of the batch PSBs and DBDs
- rr = P3 for the transactions prefix.

Declaration of the DBD in use:

DBD names	DATASETS names
PACDTDxx	PAC7TDxx
PACDTExx	PAC7TExx
PACDTGxx	PAC7TGxx
PACDTUxx	PAC7TUxx
PACDTVxx	PAC7TVxx
PACDTZxx	PAC7TZxx
PACDT1xx	PAC7T1xx
PACDTBxx	PAC7TBxx

```
DATABASE ACCESS=UP,DBD=(PACDTDyy, ... etc ...)
```

Declaration of Pactables transaction codes: (Conversational transactions)

```
a) APPLCTN  PSB=BVPPxxx
TRANSACT CODE=rrCxxx,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,
         PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
         MSGTYPE=(SNGLSEG,RESPONSE,1),SPA=(150,DASD)
```

with the following values for xxx:

```
         xxx=500   xxx=510   xxx=520   xxx=530   xxx=540
         xxx=550   xxx=560   xxx=570   xxx=580   xxx=590
         xxx=599   xxx=600   xxx=610   xxx=620   xxx=LNK
```

```
b) APPLCTN  PSB=BVPP501
TRANSACT CODE=tttttttt,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,
         PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
         MSGTYPE=(SNGLSEG,RESPONSE,1),SPA=(150,DASD)
```

with 'tttttttt': the transaction code you choose to access Pactables.

```
c) APPLCTN  PSB=BVPP601
TRANSACT CODE=mmmmmmmm,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,
         PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
         MSGTYPE=(SNGLSEG,RESPONSE,1),SPA=(150,DASD)
```

With 'mmmmmmmm': the transaction code you choose to access the Administration part of the Pactables function.

Moreover TP Pactables in character mode bypasses the MFS system and the transaction is programmed to be in SNGLSEG mode. As a result, the messages' buffers must be able to receive 3,500 bytes: RECLNG parameter of the MSGQUEU Macro or OUTBUF parameter of the TERMINAL Macro if static terminals are declared.

Notes:

It is highly recommended to use Pactables transactions in RESPONSE mode.

SPAs must not be defined as 'FIXED'.

BMP declaration

```
APPLCTN  PSB=PTA250xx,SCHDTYP=PARALLEL
APPLCTN  PSB=PTA300xx,SCHDTYP=PARALLEL
```

Note: the sources of the parameters for IMS generation are located on the cartridge in the hlq.SBVPINST file.

Chapter 4. The batch procedures

Introduction

Batch processing with Pactables is divided into various procedures. The procedures likely to be used are described in the following chapters with their specific execution conditions.

For each procedure, there is:

- A general presentation including:
  - the introduction,
  - the execution condition(s),
  - the action(s) to be performed in case of abnormal execution.
- A description of user input, processing executed, and results, plus - if needed - specific recommendations on use.
- A description of steps:
  - notations or parameters used,
  - list of the files used (temporary or permanent),
  - return codes for each step
- JCL lines.

Classification of procedures

The batch procedures are the following:

- Pactables file initialization (INTA)
- Transfer of TD-TV databases onto VSAM files (BVTA)
- Table generation (GETT)
- Table Update (UPTA)
- Table printing (PRTA)
- Table importation (IMTA)
- Table reorganization (RETA)
- Table backup (SVTA)
- Pactables database migration (TCTA)
- Table restoration (RSTA)
- Printing of table description lists (LDTA)
- Update of user parameters (PMTA)
- Extraction of data (EXTA)

- Direct reading of tables (TUTA)
- Table description comparison (CDT1, CDT2)
- Table extraction for update (CVTA).

Note

Pactables does not provide a journal of update transactions.

Abnormal executions

Generally, all batch programs may end abnormally. Specifically, I/O errors on the system's database provoke an abnormal end via an OC7 ABEND accompanied by an error message on the SYSOUT file.

In case of an abend, the user must first locate the error message, displayed as follows:

```
********** END OF RUN DUE TO AN INPUT-OUTPUT ERROR,
           PROVOKED ABEND
-------------------------------------------------
  FILE : ff  OPER : oo  KEY : kkkkkkkkkkkkkkkkkkk
  NAME OF DATABASE             :  DBDNAME
  NAME OF SEGMENT              :  SEGMname
  RETURN CODE                  :  rc
  PROCESSING OPTION            :  PROCOPT
-------------------------------------------------
```

Library code, Session number, Date of compile, Program code.

If the message is absent and the ABEND is caused directly by the VA Pac system, contact the IBM Technical Support and save all listings that may be needed to analyse the problem.

INTA - Table initialization

INTA - Introduction

This procedure initializes the files which contain the descriptions and contents of tables.

Note:

The purpose of this procedure is to physically initialize new files. It may not be used to initialize new tables in already defined files (refer to Chapter 'The Batch procedures', Subchapter 'GETT - Table generation' for more details on the Table initialization procedure).

EXECUTION CONDITION

Since the TV and TD files are updated by this procedure, the access to on-line use must be closed.

INTA - User input

Pos.	Len.	Value	Meaning
1	36		Installation label
37	1		Language version parameter:
		E	English
		F	French
38	1		Not used
39	12		Function keys assignments
51	4	cccc	Security system class
55	1		Security system type
		blank	No security system
		R	RACF
		S	TOP SECRET
56	2	nn	Number of lines per printout page
58	1		Type of resource control
		blank	Def tables resource security system
		P	Def of resources in VA Pac
59	1		Lock of user's code
		blank	Other user's code authorized
		N	Other user's code unauthorized

INTA - Description of steps

Input recognition: PTU001

Definition of files: IDCAMS

Code	Physical name	Type	Label
PAC7TD	&INDUV..&BASE.TD		Table-description file

Code	Physical name	Type	Label
PAC7TV	&INDUV..&BASE.TV		Table-contents file
PAC7TU	&INDUV..&BASE.TU		Table-contents index

Initialization of files: PTAINI

Code	Physical name	Type	Label
PAC7MD	&&INTAMB	Input	Input file
PAC7TD$SUG	&INDUV..&BASE.TD	Output	Table-description file
PAC7TV$SUG	&INDUV..&BASE.TV	Output	Table-contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Output	Table-contents index
PAC7ED		Report	Initialization review

INTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*              - INITIALIZATION OF TABLE MANAGEMENT FILE -          *
//*********************************************************************
//BVPINTA  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         VA PACTABLES FILE INDEX
//         INDSN='$INDSN',         INDEX OF NON-VSAM FILES
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SPAMB='(TRK,(1,1),RLSE)', MB FILE SPACE
//         OUT='$OUT',             OUTPUT CLASS
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY  EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//PAC7MB  DD DSN=&&INTAMB,DISP=(,PASS),UNIT=$UWK,
//           DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE   DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DF&BASE.TD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(DF&BASE.TU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(DF&BASE.TV),DISP=SHR
//*-------------------------------------------------------------
//PTAINI  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAINI,PTAINI$SUG,&BUF,
```

```
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MD   DD DSN=&&INTAMB,DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7ED   DD SYSOUT=&OUT
//*
```

BVTA - TD-TV databases on VSAM files

BVTA - Introduction

This utility function executes the BVPTABVS program which copies the TD and TV DL/1 databases onto VSAM files.

Execution condition

None since the database is not updated.

BVTA - Description of steps

Definition of files: IDCAMS

Code	Physical name	Type	Label
PAC7TS	&INDUV..&BASE.TS		Tables descriptions file
PAC7TW	&INDUV..&BASE.TW		Tables contents file

Check of VSAM files: IDCAMS

Copy of the datatabses onto VSAM files: PTABVS

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Tables descriptions file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Tables contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Tables contents index
PAC7TS	&INDUV..&BASE.TS	Output	Tables descriptions file
PAC7TW	&INDUV..&BASE.TW	Output	Tables contents file

BVTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*       - COPY OF TD-TV DATABASES ONTO TS-TW VSAM FILES-            *
//*********************************************************************
//BVPBVTA  PROC BASE=$BASE,        CODE OF VA PACTABLES DATABASE
//         INDUV='$INDUV',         USER VSAM FILES INDEX
//         INDSN='$INDSN',         SYSTEM FILE INDEX
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//         OUT='$OUT',             OUTPUT CLASS
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(DF&BASE.TW),DISP=SHR
//      DD DSN=&INDSN..BVPSY(DF&BASE.TS),DISP=SHR
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTABVS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTABVS,PTABVS$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
```

```
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7TS   DD DSN=&INDUV..&BASE.TS,DISP=SHR
//PAC7TW   DD DSN=&INDUV..&BASE.TW,DISP=SHR
//*
```

GETT - Table generation

GETT - Introduction

This procedure updates the tables descriptions file using the tables descriptions extracted from the VisualAge Pacbase Database, and initializes the generated tables in the Tables Contents file.

Execution condition

This procedure must be preceded by the Extraction procedure of the VisualAge Pacbase system (GETD or GETA), whose output file contains the extracted tables descriptions used as input to the GETT procedure.

The TD and TV files being updated by this procedure, access to on-line use must be closed if this procedure is executed in a DLI mode.

NOTE about the platforms where the disk space allocated to the files is fixed:

When a very large update (in terms of number of transactions) is run, it may be necessary to precede the execution of this procedure by a backup and a reload in order to increase or physically reorganize the files and make all the initially provided free space available.

User input

Result of GETD or GETA extraction.

GETT - Description of steps

Check of VSAM files: IDCAMS

Update of table files: PTA250

Code	Physical name	Type	Label
PAC7MD	&MD	Input	Transaction file (GETD or GETA output)
PAC7TD$SUG	&INDUV..&BASE.TD	Input Output	Tables descriptions file
PAC7TV$SUG	&INDUV..&BASE.TV	Input Output	Tables contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input Output	Tables contents index
PAC7TK	&&DE	Output	Output file
PAC7ET		Report	Input/output errors on files
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Printing of descriptions: PTA290

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Tables descriptions file
PAC7TE	&&DE	Input	Print request
PAC7ID		Report	Printout of descriptions

GETT - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*               - GENERATION OF TABLES (PACTABLES SITE) -           *
//*********************************************************************
//BVPGETT  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         VA PACTABLES FILE'S INDEX
//         INDSN='$INDSN',         NO VSAM SYSTEM FILES'S INDEX
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',        SORT LIBRARY
//         OUT='$OUT',             OUTPUT CLASS
//         OUTL=$OUT,              OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         MD='&&MD',              GENERATED DESCRITPION DSN
//         SPAWK='(TRK,(50,10),RLSE)',  SPACE OF WORK FILES
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
```

```
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         IN=,OUT1=,OPT=N,DIRCA=000,STIMER=,PARDLI=1,
//         CPUTIME=,NBA=,OBA=,IMSID=$IMSID,AGN=,
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTA250  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(BMP,BVPTA250,PTA250$SUG,&IN,&OUT1,
//        &OPT&SPIE&TEST&DIRCA,&PRLD,&STIMER,&CKPTID,
//        &PARDLI,&CPUTIME,&NBA,&OBA,&IMSID,&AGN)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7MD   DD DSN=&MD,DISP=SHR
//PAC7TK   DD DSN=&&DE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),
//            SPACE=&SPAWK
//PAC7ET   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA290  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA290,PTA290$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
```

```
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&&DE,DISP=(OLD,PASS)
//PAC7ID   DD SYSOUT=&OUTL
//*
```

UPTA - Table update

UPTA - Introduction

This procedure executes a batch update of the tables, and prints the updated tables.

Execution condition

Since this procedure updates the TV and TD Table files, access to on-line use must be closed if this procedure is executed in a DLIBATCH.

NOTE about the platforms where the disk space allocated to the files is fixed:

When a very large update is run (in terms of the number of transactions), it may be necessary precede the execution of this procedure by a backup and a reload in order to increase or physically reorganize the TV file to make all the initially provided free space available.

Important note

An alternative version of the update program, BVPTA302, is available from Pactables 2.0 onwards.

During updates, the BVPTA300 program may call the user validation routines in order to perform additional controls. The default generation option for these routines is 'without century management'.

From Release 2.0 onwards, the user validation routines are generated with the century-management option. The new program, BVPTA302, must therefore be renamed and used instead of the BVPTA300 program.

In all cases, ALL the user validation routines should be generated with the same century-management option.

UPTA - User input

- One '*'-type line per user:

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

- One 'A'-type line per table to be updated:

Pos.	Len.	Value	Meaning
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Historical account date
17	1		Not used
18	1		Sub-system number
		' '	No sub-system specified
		1 to 0	Sub-system number
19	1		Data delimiter
		' '	Considered as '/' by default

- 'V'-type lines to update table data:

Pos.	Len.	Value	Meaning
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	1	'V'	Line code

Pos.	Len.	Value	Meaning
3	1		Continuation line
		' '	First data line
		'-'	Item data continuation
4	77		Table data separated by the delimiter indicated on the 'A'-type line

UPTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Table update: PTA300

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Tables descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameters file
PAC7TV$SUG	&INDUV..&BASE.TV	Input Output	Tables contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input Output	Tables contents index
PAC7MS	&&UPTAMB	Input	Update transactions
PAC7DE	&&TABLE	Output	Print requests (lrecl=80)
PAC7ET		Report	Transaction review
PAC7MT	&&MVT300		Formatted transactions

Formatting for printing: PTA350

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7DE	&&TABLE	Input	Print request
PAC7ET	&&SPOOL	Output	Print file
PAC7EX		Report	Statistics on printing

Printing of tables: PTA360

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Tables descriptions file
PAC7ET	&&SPOOL	Input	Print file
PAC7EY		Report	Printing of tables
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

UPTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                 - TABLE UPDATE -                                  *
//*********************************************************************
//BVPUPTA  PROC BASE='$BASE',      CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',         INDEX OF VA PACTABLES SYSTEM
//         INDSN='$INDSN',         INDEX OF NON-VSAM FILES
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',        SORT LIBRARY
//         OUT='$OUT',             OUTPUT CLASS
//         OUTL=$OUT,              OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         CYL=3,                  SPACE SORTWK
//         SPAWK='(TRK,(50,10),RLSE)',  SPACE OF WORK FILES
//         SPAED='(TRK,(150,30),RLSE)', SPACE OF SPOOL FILES
//         SPAMB='(TRK,(5,1),RLSE)',    SPACE OF TRANSACTION
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         IN=,OUT1=,OPT=N,DIRCA=000,STIMER=,PARDLI=1,
//         CPUTIME=,NBA=,OBA=,IMSID=$IMSID,AGN=,
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&UPTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
```

```
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTA300  EXEC PGM=DFSRRC00,REGION=1536K,
//        PARM=(BMP,BVPTA300,PTA300$SUG,&IN,&OUT1,
//        &OPT&SPIE&TEST&DIRCA,&PRLD,&STIMER,&CKPTID,
//        &PARDLI,&CPUTIME,&NBA,&OBA,&IMSID,&AGN)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7MS   DD DSN=&&UPTAMB,DISP=(OLD,PASS)
//PAC7DE   DD DSN=&&TABLE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAWK
//PAC7MT   DD DSN=&&MVT300,DISP=(NEW,DELETE),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=95,BLKSIZE=6175),
//            SPACE=&SPAWK
//PAC7ET   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA350  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA350,PTA350$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
```

```
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7DE   DD DSN=&&TABLE,DISP=(OLD,DELETE)
//PAC7ET   DD DSN=&&SPOOL,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=160,BLKSIZE=1600),
//            SPACE=&SPAED
//PAC7EX   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA360  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA360,PTA360$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
```

```
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7ET   DD DSN=&&SPOOL,DISP=(OLD,DELETE)
//PAC7EY   DD SYSOUT=&OUTL
//*
```

PRTA - Table printing

PRTA - Introduction

This procedure performs a batch printing of tables.

Execution condition

This procedure reads the Pactables files ; it can be executed even if access to on-line use remains open.

Note

Users may also submit the PRTA procedure in on-line mode: refer to the Pactables Reference Manual for more details on batch printing submission.

PRTA - User input

- One '*'-type line per user:

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

- One 'A'-type line per table to be printed:

Pos.	Len.	Value	Meaning
1	1		Action code
		'E'	Table printing
		'H'	List of historical accounts
		'L'	List of the tables
		'S'	List of sub-schemas and sub-systems
		'X'	Table contents with historical accounts
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Historical account date or date of the reference description (if transaction code = 'X')
17	1		Sub-schema selection

Pos.	Len.	Value	Meaning
		blank	No sub-schema selection
		1 to 0	Selected sub-schema number
18	1		Sub-system selection
		blank	No sub-system selection
		1 to 0	Selected sub-system number
19	1		Print option of the key's data elements
		blank	Printing of concatenated data elements
		'O'	Printing of separated data elements

PRTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Extraction of tables for printing: PTA320

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents File
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7CA	&&PRTAMB	Input	Update transactions
PAC7DE	&&TABLE (LRECL=80)	Output	Print requests
PAC7XE		Report	Transaction review

Preparation for printing: PTA350

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7DE	&&TABLE	Input	Print requests
PAC7ET	&&SPOOL	Output	Print file
PAC7EX		Report	Statistics on printing

Printing of tables: PTA360

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7ET	&&SPOOL	Input	Print file
PAC7EY		Report	Printing of tables
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

PRTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                  - PRINTING OF TABLES -                           *
//*********************************************************************
//BVPPRTA  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',         INDEX OF VA PACTABLES SYSTEM
//         INDSN='$INDSN',         INDEX OF NON-VSAM FILES
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',        SORT LIBRARY
//         OUT='$OUT',             OUTPUT CLASS
//         OUTL=$OUT,              OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         CYL=3,                  SPACE SORTWK
//         SPAWK='(TRK,(50,10),RLSE)',  SPACE OF WORK FILES
//         SPAED='(TRK,(150,30),RLSE)', SPACE OF PRINT FILES
//         SPAMB='(TRK,(5,1),RLSE)',    SPACE OF TRANSACTION
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&PRTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
```

```
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTA320  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA320,PTA320$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7CA   DD DSN=&&PRTAMB,DISP=(OLD,DELETE)
//PAC7DE   DD DSN=&&TABLE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAWK
//PAC7XE   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA350  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA350,PTA350$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
```

```
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7DE   DD DSN=&&TABLE,DISP=(OLD,DELETE)
//PAC7ET   DD DSN=&&SPOOL,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=160,BLKSIZE=1600),
//            SPACE=&SPAED
//PAC7EX   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA360  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTA360,PTA360$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT   DD SYSOUT=&OUT
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7ET   DD DSN=&&SPOOL,DISP=(OLD,DELETE)
//PAC7EY   DD SYSOUT=&OUTL
//*
```

IMTA - Table import

IMTA - Introduction

This procedure imports external tables into the existing Pactables files.

You must first enter the description of the Table you want to import into the VA Pac Database, then generate this description (GETA/GETT procedures).

Once you have performed these operations, you can import the external table via the IMTA procedure.

The IMTA input format of the Table to be imported is a sequential file which contains one record per table item, whose contents corresponds to the description entered in the VA Pac Database (input format).

The length of this file record is 999 characters (maximum length of a table item).

Execution condition

Since this procedure updates the TV Table file, the files must be closed to on-line use except for equipment allowing batch/TP concurrence.

NOTE for platforms where the disk space allocated to the files is fixed:

If the table to be imported is large, it may be necessary to precede the execution of the procedure by a backup and reload in order to increase the size of the TV file or physically reorganize this file so as to make all the initially provided free space available.

Restriction

The procedure allows you to import only one table per execution.

Important note

An alternative version of the update program, BVPTA312, is shipped with Pactables V2.0 and onwards.

During updates, the BVPTA310 program may call user check routines in order to perform additional checks. The default generation option for these routines is 'without century management'.

With the V2.0 onwards, if the user check routines are generated with the century-management option, the new program, BVPTA312, must therefore be renamed and used instead of the BVPTA310 program.

In all cases, ALL the user check routines should be generated with the same century-management option.

IMTA - User input

- One '*'-type line per user:

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

- One 'A'-type line per table to be imported:

Pos.	Len.	Value	Meaning
2	1	'A'	Line code
3	6	tttttt	Number of the table to be imported
9	8	DDMMCCYY	Table date (optional)

IMTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Table check and update: PTA310

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7TV$SUG	&INDUV..&BASE.TV	Input output	Table contents files
PAC7TU$SUG	&INDUV..&BASE.TU	Input output	Table contents index
PAC7MV	&&IMTAMB	Input	Request transactions
PAC7NK	&TABF (LRECL=999)	Input	External table file
PAC7DE	&&TABLE (LRECL=80)	Output	Print requests
PAC7ET		Report	Execution report

Formatting of printout: PTA350

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents index
PAC7DE	&&TABLE	Input	Print requests
PAC7ET	&&SPOOL	Output	Print file
PAC7EX		Report	Printing statistics

Printing: PTA360

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7ET	&&SPOOL	Input	Print file
PAC7EY		Report	Table printout
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

IMTA - Execution JCL

```
//*************************************************************************
//*    VA PACTABLES 3.5                                                   *
//*                    - TABLE IMPORT -                                   *
//*************************************************************************
//BVPIMTA  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         VA PACTABLES FILE'S INDEX
//         INDSV='$INDSV',         VA PACTABLES SYSTEM FILE INDEX
//         INDSN='$INDSN',         NON-VSAM FILE INDEX
//         TABF=,                  DSN OF IMPORT FILE
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',        SORT LIBRARY
//         OUT='$OUT',             OUTPUT CLASS
//         OUTL=$OUT,              OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         CYL=3,                  SORTWK SPACE
//         SPAWK='(TRK,(50,10),RLSE)',  SPACE OF WORK FILES
//         SPAED='(TRK,(150,30),RLSE)', SPACE OF PRINT FILES
//         SPAMB='(TRK,(5,1))',    SPACE OF TRANSACTION FILES
//             RESLIB='$RESLIB',       IMS RESLIB
//             PROCLIB='$PRCLIB',      IMS PROCLIB
//             DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//             PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//             BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
```

```
//             CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&IMTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTA310  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA310,PTA310$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7MV   DD DSN=&&IMTAMB,DISP=(OLD,PASS)
//PAC7NK   DD DSN=&TABF,DISP=OLD
```

```
//PAC7DE   DD DSN=&&TABLE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAWK
//PAC7ET   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA350  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA350,PTA350$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7DE   DD DSN=&&TABLE,DISP=(OLD,DELETE)
//PAC7ET   DD DSN=&&SPOOL,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=160,BLKSIZE=1600),
//            SPACE=&SPAED
//PAC7EX   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA360  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA360,PTA360$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
```

```
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT   DD SYSOUT=&OUT
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7ET   DD DSN=&&SPOOL,DISP=(OLD,DELETE)
//PAC7EY   DD SYSOUT=&OUTL
//*
```

RETA - Table reorganization

RETA - Introduction

From the Pactables Database, this procedure rebuilds the backup file containing the new table descriptions and contents files, reorganized images of the initial TD and TV files.

RETA deletes the records that were logically deleted during update by reorganizing the historical accounts of the files according to the user's requests (see the Pactables Reference Manual). The records that were logically deleted can be kept by option.

For user programs written in Cobol II, RETA assigns a sign + to positive signed numeric data (not available in previous releases).

Execution condition

To ensure the consistency of the reorganized database, files must be closed to on-line use.

RETA - User input

- One '*'-type line identifying the Pactables Manager :

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

- One 'A'-type line per historical account to keep or delete:

Pos.	Len.	Value	Meaning
1	1		Action code
		'S'	Historical account to purge
		'G'	Historical account to keep
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Historical account date
19	1	' '	Option
			- when the action code is equal to 'G', the historical account whose date is equal to the date specified is kept. If there is no date, all historical accounts are kept.
			- When the action code is equal to 'S', the historical account whose date is equal to the date specified is purged.
		'<'	- When the action code is equal to 'G', the historical accounts whose dates are strictly earlier than the date specified are kept.
			- When the action code is equal to 'S', all historical accounts whose dates are strictly earlier than the date specified are purged.
		'>'	- When the action code is equal to 'G', all historical accounts whose dates are later than or equal to the date specified are kept.
			- When the action code is equal to 'S', all historical accounts whose dates are later than or equal to the date specified are purged.

The action codes 'G' and 'S' are exclusive.

For more details, see the Pactables Reference Manual.

RETA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Reorganization of table contents: PTA400

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file

Code	Physical name	Type	Label
PAC7TE$SUG	&INDSV..BVPTE	Input	Error-message file
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7DR	&&RETAMB	Input	Reorganization requests
PAC7TX	&&TXREO (LRECL=1063)	Output	Reorganized contents file
PAC7DE	&&DEREO (LRECL=80)	Output	Reorganized table list file
PAC7IR		Report	Transaction report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Note:

The PAC7DE file (reorganized table list file) whose description contains print requests, may be kept. Once the reorganization is complete, it can be used as input for the PRTA procedure applied to the reorganized files, thus enabling the printing of all the tables that were kept, in order to check the correct execution of the reorganization.

Return codes:

- 0: No error detected.
- 4: Error on an 'A' line.

Validation of table contents: PTA410

Code	Physical name	Type	Label
PAC7MB	&&RETAMB	Input	Reorganization requests
PAC7TX	&&TXREO	Input	Reorganized contents file
PAC7TW	&&TWREO	Output	Validated contents file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Reorganization of table descriptions: PTA420

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7DE	&&DEREO	Input	Reorganized table list file
PAC7TS	&&TSREO	Output	Reorganized tables descriptions file
PAC7ML	&&ME	Output	Table descriptions print request
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Note

The PAC7ML file (Tables descriptions print request) must be kept and used as input tp the LDTA procedure, to produce a printout of the table descriptions that were kept, in order to check the correct execution of the reorganization.

Building of backup file: PTA430

Code	Physical name	Type	Label
PAC7TW	&&TWREO	Input	Validated contents file
PAC7TS	&&TSREO	Input	Reorganized descriptions file
PAC7TC	&INDUN..&BASE.TC(+1)	Output	Backup file resulting from reorganization

TG file backup: PTASVG

Code	Physical name	Type	Label
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User-parameter file
PAC7TC	&INDUN..&BASE.TC(+1)	Output	Table backup

RETA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                  - REORGANIZATION OF TABLES -                     *
//*********************************************************************
//BVPRETA  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',         INDEX OF VA PACTABLES SYSTEM
//         INDSN='$INDSN',         INDEX OF NON-VSAM FILES
```

```
//         INDUN='$INDUN',         NON-VSAM USER FILE INDEX
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',        SORT LIBRARY
//         OUT='$OUT',             OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         CYL='(3,1)',            SORTWORK SPACE
//         VOLS='SER=$VOLUN',      BACKUP VOLUME
//         DSCB='$DSCB',           DSCB MODEL
//         SPAWK='(TRK,(50,10),RLSE)', SPACE OF WORK FILES
//         SPAMB='(TRK,(5,1),RLSE)',   SPACE OF TRANSACTIONS
//         SPATC='(TRK,(150,10),RLSE)', SPACE OF BACKUP FILE
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//STEPLIB  DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&RETAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTA400  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA400,PTA400$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
```

```
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7DR   DD DSN=&&RETAMB,DISP=(OLD,PASS)
//PAC7DE   DD DSN=&&DEREO,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
//            SPACE=&SPAWK
//PAC7TX   DD DSN=&&TXREO,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=VB,LRECL=1063,BLKSIZE=10630),
//            SPACE=&SPAWK
//PAC7IR   DD SYSOUT=&OUT
//*-------------------------------------------------------------
//PTA410   EXEC PGM=BVPTA410
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&RETAMB,DISP=(OLD,PASS)
//PAC7TX   DD DSN=&&TXREO,DISP=(OLD,PASS)
//PAC7TW   DD DSN=&&TWREO,DISP=(,PASS),UNIT=&UWK,
//         SPACE=&SPAWK,
//         DCB=(RECFM=VB,LRECL=1063,BLKSIZE=10630)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//*-----------------------------------------------------------
//PTA420  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA420,PTA420$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
```

```
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TS   DD DSN=&&TSREO,DISP=(NEW,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=240,BLKSIZE=2400),
//            SPACE=&SPAWK
//PAC7DE   DD DSN=&&DEREO,DISP=(OLD,PASS)
//PAC7ML   DD DSN=&&ME,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
//            SPACE=&SPAWK
//*-------------------------------------------------------------
//PTA430   EXEC PGM=BVPTA430,COND=(8,LE,PTA400)
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//STEPLIB  DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7TS   DD DSN=&&TSREO,DISP=(OLD,PASS)
//PAC7TW   DD DSN=&&TWREO,DISP=(OLD,PASS)
//PAC7TC   DD DSN=&INDUN..&BASE.TC(+1),
//         UNIT=&UWK,VOL=&VOLS,
//         DISP=(NEW,CATLG,DELETE),
//         SPACE=&SPATC,
//         DCB=(&DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*-------------------------------------------------------------
//PTASVG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTASVG,PTASVG$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(8,LE,PTA400)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
```

```
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(+1),
//         UNIT=&UWK,VOL=&VOLS,
//         DISP=MOD,
//         DCB=(&DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//*
```

SVTA - Backup

SVTA - Introduction

The SVTA procedure performs a backup of the Tables descriptions and contents, and a backup of the user parameters in a single sequential file (TC).

Execution condition

In order to keep a consistency in the data, it is recommended to close the files to on-line use.

User input

None.

SVTA - Description of steps

TD backup: PTASVT

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TC	&INDUN..&BASE.TC(+1)	Output	Table backup

TV backup: PTASVV

Code	Physical name	Type	Label
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7TC	&INDUN..&BASE.TC(+1)	Output	Table backup

TG backup: PTASVG

Code	Physical name	Type	Label
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7TC	&INDUV..&BASE.TC(+1)	Output	Table backup

SVTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*          - BACKUP OF THE PACTABLES DATABASE (TD, TV, TG) -        *
//*********************************************************************
//BVPSVTA  PROC BASE=$BASE,        CODE OF VA PACTABLES DATABASE
//         INDUV='$INDUV',         USER VSAM FILES INDEX
//         INDUN='$INDUN',         INDEX OF NON VSAM FILES
//         INDSN='$INDSN',         SYSTEM FILE INDEX
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//         SPATC='(TRK,(30,3),RLSE)', BACKUP-FILE SPACE
//         VOLS='SER=$VOLUN',       BACKUP-FILE VOLUME
//         UNITS='$UNITUN',         BACKUP-FILE UNIT
//         DSCB='$DSCB',           DSCB MODEL
//         OUT='$OUT',             OUTPUT CLASS
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//*-------------------------------------------------------------
//PTASVT  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTASVT,PTASVT$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
```

```
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=(&DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//*-------------------------------------------------------------
//PTASVV  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTASVV,PTASVV$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(+1),DISP=MOD,
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=(&DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//*-------------------------------------------------------------
//PTASVG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTASVG,PTASVG$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
```

```
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(+1),DISP=MOD,
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=(&DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//*
```

TCTA - Pactables transfer from another platform

TCTA - Introduction

The purpose of this procedure is to retrieve Pactables Databases from other platforms (source platforms) in order to adapt them to your environment.

The Database backup is sorted according to the format of the target platform (ASCII or EBCDIC).

If the version of the source site is the same as that of the target site, the actions to be performed are the following ones:

- Backup on the source site (SVTA procedure)
- Transfer of the TC file produced by SVTA onto the target platform,
- Retrieval of the file on the target platform (TCTA procedure),
- Restoration of the database (RSTA procedure), with, in input, the TC file built by the preceding step.

If, on the contrary, the source site is of an older version, and that the version requires a retrieval, the TC backup must be retrieved in the new format ON THE SOURCE SITE before being transferred onto the target environment.

Execution condition

None. But carefully read the note below.

User input

None.

Notes

1. Backup transfer

   Transferring the TC backup from the source site to the mainframe host where Pactables is installed is the user's responsibility. The file contains data (DATA) which must be converted in the EBCDIC format.

2. Disk space

   The TCTA procedure, described thereafter, consists mainly of sorting the TC backup according to an EBCDIC sequence. The sort is performed in three distinct steps, so as to minimize the disk space required. However, the procedure requires between 4 to 4.5 times the equivalent of the original file's size.

3. Initial and result files

   The TC backup input file, coming from a different platform, is specified in the TCTA procedure by its DSNAME, by setting a value to the SAVIN parameter. As a default, generation 0 of the Data-group corresponding to the backup of the Pactables database is used.

   As a default, the TC output file, sorted according to the EBCDIC sequence, corresponds to the generation +1 of the Pactables database backup. This file may be retrieved by performing an override on the PTATC2.PAC7TC procedure (see the provided execution test JCL).

TCTA - Description of steps

TC backup split: PTATC1

Code	Physical name	Type	Label
PAC7TC	&SAVIN	Input	Input backup file
PAC7SD	&&PAC7SD	Output	Sequential image of table descriptions
PAC7SV	&&PAC7SV	Output	Sequential image of table contents
PAC7SG	&&PAC7SG	Output	Parameter sequential image

Table-description sort: PTATCD

Code	Physical name	Type	Label
PAC7SD	&&PAC7SD	Input	Sequential image of table descriptions
PAC7AD	&&PAC7AD	Output	Sorted table descriptions

Table-contents sort: PTATCV

Code	Physical name	Type	Label
PAC7SV	&&PAC7SV	Input	Sequential image of table contents
PAC7AV	&&PAC7AV	Output	Sorted tables contents

User-parameter sort: PTATCG

Code	Physical name	Type	Label
PAC7SG	&&PAC7SG	Input	Sequential image of parameters
PAC7AG	&&PAC7AG	Output	Sorted user parameters

Reconstitution of the TC backup: PTATC2

Code	Physical name	Type	Label
PAC7AD	&&PAC7AD	Input	Sequential image of table descriptions
PAC7AV	&&PAC7AV	Input	Sequential image of contents
PAC7AG	&&PAC7AG	Input	Sequential image of parameters
PAC7TC	&INDUN..&BASE.TC(+1)	Output	TC backup in EBCDIC format

TCTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*      ----  TRANSFER OF A PACTABLES DATABASE ---                   *
//*********************************************************************
//BVPTCTA  PROC BASE=$BASE,      CODE OF PACTABLES DATABASE
//         INDUN='$INDUN',       BACKUP FILES' INDEX
//*:       VSAMCAT='$VCAT',      USER VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         VOLS='SER=$VOLUN',    BACKUP VOLUME
//         UWK=$UWK,             WORK UNIT
//         OUT=$OUT,             OUTPUT CLASS
//         DSCB='$DSCB',           DSCB MODEL
//         SAVIN='$INDUN..$BASE.TC(0)',  BACKUP FILE
//         SPASD='(TRK,(15,5),RLSE)',  'SD' FILE SPACE
//         SPASV='(TRK,(15,5),RLSE)',  'SV' FILE SPACE
//         SPASG='(TRK,(15,5),RLSE)',  'SG' FILE SPACE
//         CYL='(3,1)'                SORTWORK SPACE
```

```
//*************************************************************
//PTATC1   EXEC PGM=BVPTATC1
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7TC   DD DSN=&SAVIN,DISP=SHR
//PAC7SD   DD DSN=&&PAC7SD,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPASD,
//         DCB=(RECFM=FB,LRECL=244,BLKSIZE=24400)
//PAC7SV   DD DSN=&&PAC7SV,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPASV,
//         DCB=(RECFM=FB,LRECL=1063,BLKSIZE=10630)
//PAC7SG   DD DSN=&&PAC7SG,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPASG,
//         DCB=(RECFM=FB,LRECL=89,BLKSIZE=8900)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*-------------------------------------------------------------
//PTATCD   EXEC PGM=BVPTATCD
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7SD   DD DSN=&&PAC7SD,DISP=SHR
//PAC7AD   DD DSN=&&PAC7AD,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPASD,
//         DCB=(RECFM=FB,LRECL=244,BLKSIZE=24400)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*-------------------------------------------------------------
//PTATCV   EXEC PGM=BVPTATCV
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7SV   DD DSN=&&PAC7SV,DISP=SHR
//PAC7AV   DD DSN=&&PAC7AV,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPASV,
//         DCB=(RECFM=FB,LRECL=1063,BLKSIZE=10630)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*-------------------------------------------------------------
//PTATCG   EXEC PGM=BVPTATCG
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7SG   DD DSN=&&PAC7SG,DISP=SHR
//PAC7AG   DD DSN=&&PAC7AG,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPASG,
//         DCB=(RECFM=FB,LRECL=89,BLKSIZE=8900)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

```
//*-------------------------------------------------------------
//PTATC2   EXEC PGM=BVPTATC2
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7AD   DD DSN=&&PAC7AD,DISP=SHR
//PAC7AV   DD DSN=&&PAC7AV,DISP=SHR
//PAC7AG   DD DSN=&&PAC7AG,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(+1),DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,VOL=&VOLS,
//         SPACE=(TRK,(15,5),RLSE),
//         DCB=(&DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

RSTA - Restoration

RSTA - Introduction

The RSTA procedure is used to restore the descriptions and contents of tables, as well as the user parameters, from the sequential image obtained by the SVTA backup procedure.

Execution condition

In order to keep a consistency in the data, access to on-line use must be closed.

NOTE about the platforms where the disk space allocated to the files is fixed:

As this procedure reloads the files, it is recommended to consider beforehand the estimated evolution of the files and re-adjust their size accordingly. These modifications should be made in the system parameters library.

Abnormal execution

See Chapter 'Batch procedures', Subchapter 'Abnormal Executions'.

Whatever the cause of the abend, the procedure can be restarted as it is once the problem has been solved.

User input

None.

RSTA - Description of steps

Definition of files: IDCAMS

Code	Physical name	Label
PAC7TD	&INDUV..&BASE.TD	Table descriptions file
PAC7TV	&INDUV..&BASE.TV	Table contents file
PAC7TU	&INDUV..&BASE.TU	Table contents index
PAC7TG	&INDUV..&BASE.TG	User parameter file

Restoration of TD: PTARST

Code	Physical name	Type	Label
PAC7TC	&INDUN..&BASE.TC(0)	Input	Table backup
PAC7TD$SUG	&INDUV..&BASE.TD	Output	Table descriptions file

Restoration of TV: PTARSV

Code	Physical name	Type	Label
PAC7TC	&INDUV..&BASE.TC(0)	Input	Table backup
PAC7TV$SUG	&INDUV..&BASE.TV	Output	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Output	Table contents index

Restoration of TG: PTARSG

Code	Physical name	Type	Label
PAC7TC	&INDUN..&BASE.TC(0)	Input	Table backup
PAC7TG$SUG	&INDUV..&BASE.TG	Output	User parameter file

RSTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*              - RELOADING-RESTORING TABLES (TD-TV-TG) -            *
//*********************************************************************
//BVPRSTA  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         INDEX OF VA PACTABLES FILES
//         INDUN='$INDUN',         INDEX OF BACKUP FILES
//         INDSN='$INDSN',         INDEX NON VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//         OUT='$OUT',             OUTPUT CLASS
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
```

```
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//*
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN  DD DSN=&INDSN..BVPSY(DF&BASE.TD),DISP=SHR
//       DD DSN=&INDSN..BVPSY(DF&BASE.TU),DISP=SHR
//       DD DSN=&INDSN..BVPSY(DF&BASE.TV),DISP=SHR
//       DD DSN=&INDSN..BVPSY(DF&BASE.TG),DISP=SHR
//*-------------------------------------------------------------
//PTARST  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTARST,PTARST$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(0),DISP=SHR
//*-------------------------------------------------------------
//PTARSV  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTARSV,PTARSV$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
```

```
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(0),DISP=SHR
//*--------------------------------------------------------------
//PTARSG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTARSG,PTARSG$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TC   DD DSN=&INDUN..&BASE.TC(0),DISP=SHR
//*
```

LDTA - List of table descriptions

LDTA - Introduction

This procedure prints descriptions of tables.

Execution condition

This procedure reads the TD file which can remain open to on-line use.

LDTA - User input

A 'Z'-type line per print request:

Pos.	Len.	Value	Meaning
2	1	'Z'	Line code
5	4		Print request
		'TLS '	List of table descriptions
		'TDS '	Table description
9	6	ttttt	Table number
23	8	MMDDCCYY	Historical account date

Note

The input transactions are not validated; erroneous requests are ignored.

LDTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Printing of tables descriptions: PTA290

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TE	&&LDTAMB	Input	Print request
PAC7ID		Report	Table descriptions printout

LDTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*               - TABLE DESCRIPTIONS LIST -                         *
//*********************************************************************
//BVPLDTA  PROC BASE=$BASE,        CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',         INDEX OF VA PACTABLES FILES
//         INDSN='$INDSN',         INDEX OF NON-VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTIONS F
//         OUT='$OUT',             OUTPUT CLASS

//         OUTL=$OUT,              OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
```

```
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&LDTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//*-------------------------------------------------------------
//PTA290  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA290,PTA290$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&&LDTAMB,DISP=(OLD,PASS)
//PAC7ID   DD SYSOUT=&OUTL
//*
```

PMTA - Parameter update

PMTA - Introduction

This procedure updates Pactables user codes, passwords and access authorizations as well as control cards for print requests.

When the user input contains a 'TA' line with the Database Administrator user's code, the PMTA procedure prints all the user parameters.

Execution condition

This procedure updates the TG file, which must be closed to on-line use except if the hardware in use allows Batch/TP concurrency.

PMTA - User input

- 'TA'-line: user parameter updating:

Pos.	Len.	Value	Meaning
1	1		Action code
		blank	Creation or modification
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	8	uuuuuuuu	User code
10	2	'TA'	Line code
12	8	pppppppp	Password
20	1		General access authorization
		'0'	No general access authorization
		'1'	Read-only access authorization
		'2'	Read-write authorization on tables
		'3'	Read-write authorization on user codes

- 'TC'-line: access authorizations per table:

Pos.	Len.	Value	Meaning
1	1		Action code
		blank	Creation or modification
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	8	uuuuuuuu	User code
10	2	'TC'	Line code
12	6	tttttt	Table code
18	3	nnn	Line number
21	60		Access authorizations: 20 access authorizations may be entered in this field, with, for each authorization:
	1	n	Sub-schema number

Pos.	Len.	Value	Meaning
	1	n	Sub-system number
	1	x	Authorization (0, 1 or 2) ('*' for all sub-schemas and sub-systems)

- 'TJ'-line: control cards:

Pos.	Len.	Value	Meaning
1	1		Action code
		blank	Creation or modification
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	8	uuuuuuuu	User code
10	2	'TJ'	Line code
12	6		JCL line number
		<600000	Control card in front of program
		>599999	Control card in back of program
18	69		Content of JCL line

Note

When a user code is deleted, related access authorizations and JCL lines are also deleted.

The Database must include at least one Administrator code with a level 3 access authorization. The deletion of the last Administrator code is not authorized.

PMTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

User parameters update: PTA100

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TG$SUG	&INDUV..&BASE.TG	Input Output	User parameter file

Code	Physical name	Type	Label
PAC7MV	&&PMTAMB	Input	Extraction requests
PAC7NU	&&NU	Output	Parameter printing requests
PAC7ET		Report	Printing of descriptions

Printing of user parameters: PTA120

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7NU	&&NU	Input	Print requests
PAC7ET		Report	Printing of user parameters

PMTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                - UPDATE OF USER PARAMETERS -                      *
//*********************************************************************
//BVPPMTA  PROC BASE=$BASE,        CODE OF VA PACTABLES DATABASE
//         INDUV='$INDUV',         INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',         INDEX OF VA PACTABLES SYSTEM  FILES
//         INDSN='$INDSN',         INDEX OF NON-VSAM FILES
//*:       VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',        VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         OUT='$OUT',             OUTPUT CLASS
//         OUTL=$OUT,              OUTPUT CLASS
//         UWK=$UWK,               WORK UNIT
//         SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTION FI
//         SPANU='(TRK,(5,1),RLSE)',  SPACE OF WORK FILES
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&PMTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
```

```
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//*-------------------------------------------------------------
//PTA100  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA100,PTA100$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7MV   DD DSN=&&PMTAMB,DISP=(OLD,DELETE)
//PAC7NU   DD DSN=&&NU,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPANU
//PAC7ET   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA120  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA120,PTA120$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
```

```
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7NU   DD DSN=&&NU,DISP=(OLD,DELETE)
//PAC7ET   DD SYSOUT=&OUTL
//*
```

EXTA - Table extraction

EXTA - Introduction

The EXTA procedure extracts table data in the form of batch update transactions.

Execution condition

This procedure reads the Pactables files which can remain open to on-line use.

EXTA - User input

- One '*'-type line per user:

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

- One 'A'-type line per table to extract:

Pos.	Len.	Value	Meaning
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Historical account date
17	1		Not used
18	1		Sub-system selection
		blank	No sub-system selection
		1 to 0	Number of selected sub-system
19	1		Data delimiter
		blank	'/' by default

EXTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Extraction of table data: PTA150

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7MV	&&EXTAMB	Input	Extraction requests
PAC7EX	&&EX	Output	Extracted transactions
PAC7ET		Report	Transaction review

Printing of extracted transactions: PTA160

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7EX	&&EX	Input	Extracted transactions
PAC7NU	&&MBTAB	Output	Extracted transactions
PAC7ET		Report	Printing of extracted data
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0: No delimiter in data
- 8: Delimiter in at least one table
- 12: Delimiter in all tables

EXTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                - EXTRACTION OF TABLES -                           *
//*********************************************************************
```

```
//BVPEXTA  PROC BASE=$BASE,       CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',        INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',        INDEX OF VA PACTABLES SYSTEM FILES
//         INDSN='$INDSN',        NON VSAM FILES INDEX
//*:       VSAMCAT='$VCAT',       VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',       VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',       SORT LIBRARY
//         OUT='$OUT',            OUTPUT CLASS
//         OUTL=$OUT,             OUTPUT CLASS
//         UWK=$UWK,              WORK UNIT
//         CYL='(3,1)',           SORTWORK SPACE
//         SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTION FILE
//         SPAEX='(TRK,(10,10),RLSE)', SPACE OF EXTRACTED TRANSA
//         RESLIB='$RESLIB',      IMS RESLIB
//         PROCLIB='$PRCLIB',     IMS PROCLIB
//         DBDLIB='$DBDLIB',      LIBRARY OF DBD'S
//         PSBLIB='$PSBLIB',      LIBRARY OF PSB'S
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&EXTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTA150  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA150,PTA150$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7MV   DD DSN=&&EXTAMB,DISP=(OLD,DELETE)
//PAC7EX   DD DSN=&&EX,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=120,BLKSIZE=2400),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTA160  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTA160,PTA160$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7EX   DD DSN=&&EX,DISP=(OLD,PASS)
//PAC7NU   DD DSN=&&MBTAB,DISP=(,PASS),UNIT=&UWK,
```

```
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*
```

TUTA - Direct consultation of tables

TUTA - Introduction

The TUTA procedure extracts tables in the form of tables without historical accounts and which are to be consulted.

The procedure creates two new files which contain the descriptions and contents of the selected tables. There is only one description and one version of data for each selected table.

Execution condition

This procedure recreates the AD and AV files, which must therefore be closed to on-line use. These two files are the reorganized images of TD and TV respectively.

The TUTA procedure defines both files in the second step.

TUTA - User input

- One '*'-type line :

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

- One 'A'-type line for each selected table:

Pos.	Len.	Value	Meaning
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Historical account date

When no 'A'-type line is entered, the user may use all the tables that are accessible at that time. A different date may be entered on a single 'A'-type line where no table number is indicated.

TUTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Direct consultation of tables: PTAU80

Code	Physical name	Type	Label
PAC7TD$SUG	&INDUV..&BASE.TD	Input	Table descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TV$SUG	&INDUV..&BASE.TV	Input	Table contents file
PAC7TU$SUG	&INDUV..&BASE.TU	Input	Table contents index
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7MX	&&TUTAMB	Input	Request transactions
PAC7AD	&INDUV..&BASE.AD	Output	Table descriptions file
PAC7AV	&INDUV..&BASE.AV	Output	Tables contents file
PAC7ET		Report	Transaction report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

TUTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                - DIRECT READING OF TABLES -                       *
//*********************************************************************
//BVPTUTA  PROC BASE=$BASE,   CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',    INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',    INDEX OF VA PACTABLES SYSTEL FILES
//         INDSN='$INDSN',    INDEX OF NON-VSAM FILES
//*:       SYSTCAT='$SCAT',   VSAM SYSTEM CATALOG
//*:       VSAMCAT='$VATU',   VSAM USER CATALOG
//         OUT='$OUT',        OUTPUT CLASS
//         OUTL='$OUT',       OUTPUT CLASS
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',   SORT LIBRARY
//         PSBLIB='$PSBLIB',  LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',  LIBRARY OF DBD'S
//         RESLIB='$RESLIB',  IMS RESLIB
//         PROCLIB='$PRCLIB', IMS PROCLIB
//         UWK=$UWK,          WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//INPUT   EXEC PGM=BVPTU001
```

```
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TUTAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-------------------------------------------------------------
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DF&BASE.AD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(DF&BASE.AV),DISP=SHR
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV   DD DSN=&INDUV..&BASE.TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTAU80  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAU80,PTAU80$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//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)
//PAC7TD$SUG DD DSN=&INDUV..&BASE.TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&INDUV..&BASE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDUV..&BASE.TV,DISP=SHR
//PAC7AD   DD DSN=&INDUV..&BASE.AD,DISP=SHR
//PAC7AV   DD DSN=&INDUV..&BASE.AV,DISP=SHR
//PAC7MX   DD DSN=&&TUTAMB,DISP=(OLD,DELETE)
//PAC7ET   DD SYSOUT=&OUTL
//*
```

Dispatched table management (DTM option)

CDT1-CDT2 - Table descriptions comparison

The Dispatched Table Manager is an optional utility and its use depends on a specific purchase agreement.

Table description comparison

The CDT1 procedure compares two different states of a Table description file and extracts the differences, giving an intermediate sequential file.

This file may be used to update the 'outdated' description file, called 'slave' file, (CDT2 procedure).

Execution condition

The CDT1 procedure reads the Pactables files which can therefore remain open to on-line use.

From the result of the CDT1 procedure, the CDT2 procedure updates the TD and TV files ('slave' files). These files must therefore remain closed to on-line use.

Note

The slave file (PAC7TS) used as input to the BVPTAD10 program is a VSAM file. You can use the BVTA procedure which converts the TD and TV files of the database into VSAM files.

The CDT2 procedure, on the contrary, directly updates the file from the 'slave' database.

CDT1 - User input

- One '*'-type line per user:

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

- One 'A'-type line for each selected table:

Pos.	Len.	Value	Meaning
2	1	'A'	Line code
3	6	tttttt	Table number

When a single 'A'-type line is entered without the table number, all tables descriptions are compared.

CDT1 - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Check of transactions: PTAD05

Code	Physical name	Type	Label
PAC7TD	&TDMAST	Input	Table descriptions Master file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7MV	&&CDT1MB	Input	Comparison request transactions
PAC7MX	&&MX	Output	Validated comparison request transactions
PAC7ET		Report	Transaction report

Table-description comparison and extraction: PTAD10

Code	Physical name	Type	Label
PAC7TD	&TDMAST	Input	Table description 'Master' file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file

Code	Physical name	Type	Label
PAC7TS	&TDSLAV (VSAM File)	Input	Table description 'Slave' file
PAC7MX	&&MX	Input	Validated transactions
PAC7TX	&XD	Output	Comparison result to be used as input to the CDT2 procedure
PAC7ET		Report	Extraction printout

CDT1 - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                - TABLE-DESCRIPTION COMPARISON -                   *
//*********************************************************************
//BVPCDT1  PROC BASE='$BASE',     CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',        INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',        INDEX OF VA PACTABLES SYSTEM FILES
//         INDSN='$INDSN',        NON-VSAM FILE INDEX
//*:       VSAMCAT='$VCAT',       VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',       VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',       SORT LIBRARY
//         TDMAST=,               MASTER DESCRIPTION
//         TDSLAV=,               SLAVE DESCRIPTION
//         XD='&&TX',             DSN OF EXTRACTED DESCRIPTION
//         SPAXD='(TRK,(30,10),RLSE)', SPACE OF EXTRACTED DESCR
//         SPAMB='(TRK,(5,1),RLSE)',   SPACE OF TRANSACTIONS FI
//         CYL=(3,1),             SORTWORK SPACE
//         UWK=$UWK,              WORK UNIT
//         OUT='$OUT',            OUTPUT CLASS
//         OUTL='$OUT',           OUTPUT CLASS
//         PSBLIB='$PSBLIB',      LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',      LIBRARY OF DBD'S
//         RESLIB='$RESLIB',      IMS RESLIB
//         PROCLIB='$PRCLIB',     IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//INPUT   EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&CDT1MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&TDMAST,DISP=SHR
```

```
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TS   DD DSN=&TDSLAV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTS),DISP=SHR
//*-------------------------------------------------------------
//PTAD05  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAD05,PTAD05$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUG DD DSN=&TDMAST,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7MV   DD DSN=&&CDT1MB,DISP=(OLD,DELETE)
//PAC7MX   DD DSN=&&MX,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),
//            SPACE=&SPAXD
//PAC7ET   DD SYSOUT=&OUTL
//*-------------------------------------------------------------
//PTAD10  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAD10,PTAD10$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&TDMAST,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TS   DD DSN=&TDSLAV,DISP=SHR
//PAC7MX   DD DSN=&&MX,DISP=(OLD,PASS)
//PAC7TX   DD DSN=&XD,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=240,BLKSIZE=2400),
//            SPACE=&SPAXD
//PAC7ET   DD SYSOUT=&OUTL
//*
```

CDT2 - Description of steps

Check of VSAM files: IDCAMS

Update of 'slave' files, Recognition of the file extracted by CDT1: PTAD20

Code	Physical name	Type	Label
PAC7TD	&TDSLAV	Input	Table descriptions Slave file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TX	&XD	Input	Result extracted from the comparison in the CDT1 procedure
PAC7TV$SUG	&TVSLAV	Output	Table contents file associated to the table descriptions Slave file
PAC7TU$SUG	&TUSLAV	Output	Table contents index associated to the table descriptions Slave file
PAC7ET		Report	Update report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

CDT2 - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                  - TABLE-DESCRIPTION UPDATE -                     *
//*********************************************************************
//BVPCDT2  PROC INDSV='$INDSV', INDEX OF VA PAC SYSTEM FILES
//         INDSN='$INDSN',      NON-VSAM FILE INDEX
//*:       VSAMCAT='$VCAT',     VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',     VSAM SYSTEM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         SORTLIB='$BIBT',     SORT LIBRARY
//         TDSLAV=,             SLAVE DESCRIPTION
//         TUSLAV=,             PRIMARY INDEX 'TU' OF SLAVE DESCR
//         TVSLAV=,             TABLES ASSOCIATED TO SLAVE DESCR
//         XD='&&TX',           DSN OF CDT1 EXTRACTED DESCRIPTION
//         CYL='(3,1)',         SORTWORK SPACE
//         UWK=$UWK,            WORK UNIT
//         OUT='$OUT',          OUTPUT CLASS
//         OUTL='$OUT',         OUTPUT CLASS
//         PSBLIB='$PSBLIB',    LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',    LIBRARY OF DBD'S
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&TDSLAV,DISP=SHR
//PAC7TE   DD DSN=&INDSN..BVPTE,DISP=SHR
//PAC7TU   DD DSN=&TUSLAV,DISP=SHR
//PAC7TV   DD DSN=&TVSLAV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTAD20  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAD20,PTAD20$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
```

```
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUG DD DSN=&TDSLAV,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TU$SUG DD DSN=&TUSLAV,DISP=SHR
//PAC7TV$SUG DD DSN=&TVSLAV,DISP=SHR
//PAC7TX   DD DSN=&XD,DISP=SHR
//PAC7ET   DD SYSOUT=&OUTL
//*
```

CVTA - Table contents update

The CVTA procedure extracts table contents modified on a given date, or between two given dates, and formats them as batch update transactions.

Execution condition

This procedure reads the Pactables files ; it can be executed even if the files remain open to on-line use.

CVTA - User input

- One '*'-type line per user:

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

- One 'A'-type line for each selected table:

Pos.	Len.	Value	Meaning
1	1	'S'	Transaction code
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Update date: beginning
17	2		Not used
19	1	'/'	Delimiter
20	1		Not used

Pos.	Len.	Value	Meaning
21	8	DDMMCCYY	Update date: end

When a single 'A'-type line is entered without the table number, all the modified items of all tables accessible by the user ('*'-line) can be extracted.

CVTA - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Table contents comparison: PTAV10

Code	Physical name	Type	Label
PAC7TD$SUG	&TD	Input	Table descriptions file
PAC7TE$SUG	&INDSV..BVPTE	Input	Error message file
PAC7TV$SUG	&TV	Input	Table contents file
PAC7TU$SUG	&TU	Input	Table contents index
PAC7TG$SUG	&INDUV..&BASE.TG	Input	User parameter file
PAC7MV	&&CVTAMB	Input	Comparison requests
PAC7EX	&&EX	Output	Comparison result
PAC7ET		Report	Transaction report

Extraction of update transactions: PTAV20

Code	Physical name	Type	Label
PAC7TD$SUG	&TD	Input	Table Descriptions file
PAC7EX	&&EX	Input	Comparison result
PAC7NU	&&NU	Output	Update transactions for use as input to UPTA
PAC7ET		Report	Printing of extracted transactions
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

CVTA - Execution JCL

```
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*               - EXTRACTION OF MODIFIED TABLE DATA -               *
//*********************************************************************
//BVPCVTA  PROC BASE='$BASE',   CODE OF PACTABLES DATABASE
//         INDUV='$INDUV',      INDEX OF VA PACTABLES FILES
//         INDSV='$INDSV',      INDEX OF VA PACTABLES SYSTEM FILES
//         INDSN='$INDSN',      NON-VSAM FILE INDEX
//*:       VSAMCAT='$CATV',     VSAM USER CATALOG
//*:       SYSTCAT='$SCAT',     VSAM SYSTEM CATALOG
//         SORTLIB='$BIBT',      SORT LIBRARY
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF BATCH PROGRAMS
//         TD=,                 SLAVE DESCRIPTION
//         TU=,                 'TU' PRIMARY INDEX OF SLAVE DE
//         TV=,                 TABLES ASSOCIATED TO SLAVE DES
//         SPAEX='(TRK,(30,3),RLSE)', SPACE OF EXTRACTED TRANSA
//         SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTION FILE
//         CYL='(3,1)',         SORTWORK SPACE
//         UWK=$UWK,            WORK UNIT
//         OUT='$OUT',          OUTPUT CLASS
//         OUTL='$OUT',         OUTPUT CLASS
//         PSBLIB='$PSBLIB',    LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',    LIBRARY OF DBD'S
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//INPUT   EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&CVTAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-------------------------------------------------------------
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7TD   DD DSN=&TD,DISP=SHR
//PAC7TE   DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG   DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU   DD DSN=&TU,DISP=SHR
//PAC7TV   DD DSN=&TV,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTG),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFTV),DISP=SHR
//*-------------------------------------------------------------
//PTAV10  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAV10,PTAV10$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
```

```
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7TD$SUG DD DSN=&TD,DISP=SHR
//PAC7TE$SUG DD DSN=&INDSV..BVPTE,DISP=SHR
//PAC7TG$SUG DD DSN=&INDUV..&BASE.TG,DISP=SHR
//PAC7TU$SUG DD DSN=&TU,DISP=SHR
//PAC7TV$SUG DD DSN=&TV,DISP=SHR
//PAC7MV   DD DSN=&&CVTAMB,DISP=(OLD,DELETE)
//PAC7EX   DD DSN=&&EX,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=120,BLKSIZE=2400),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*--------------------------------------------------------------
//PTAV20  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTAV20,PTAV20$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
```

```
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUG DD DSN=&TD,DISP=SHR
//PAC7EX   DD DSN=&&EX,DISP=(OLD,DELETE)
//PAC7NU   DD DSN=&&NU,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*
```

Chapter 5. Installation

Parameters

SMP/E context

VA Pacbase, DSMS and Pactables use common batch and online load modules.

These load-modules are supplied in the HBVP350 root FMID.

The components specific to VA Pacbase are supplied in the JBVP351 dependent FMID.

The components specific to DSMS are supplied in the JBVP352 dependent FMID.

The components specific to Pactables are supplied in the JBVP353 dependent FMID.

To install a dependent FMID, the HBVP350 root FMID must be simultaneously or previously installed.

If the HBVP350 root FMID is already installed, only the dependent JBVP35x FMID must be installed.

On the installation cartridge, the HBVP350 root FMID and the JBVP35x dependent FMID are systematically provided.

The RECEIVE, APPLY and ACCEPT JCLs, that are supplied, are to be used for a simultaneous installation of the two FMIDs. They must be adapted if the HBVP350 root FMID is already installed.

Preparation

The preparation to the installation process consists of three phases:

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

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

Remark for a previous SMP/E installation

The following SMP/E phase implies that the SMP/E context is empty for the product: either the product has never been been installed with SMP/E, or the files related to SMP/E have been reinitialized for the previous version, or you want to create a new SMP/E ion. environment for this version.

Otherwise, if you want to re-use the SMP/E files of theus previous version, you can execute, before phase 1, the SMP/E utility described in the Appendix at the end of this manual.

Phase 1

You will find details in the document 'Program Directory for Pactables' specific to SMP/E.

When the installation takes place in a virgin environment, the necessary SMP/E environment must be implemented and the HBVP350 and JBVP353 FMIDs must be installed with the supplied JCLs.

This stage consists of the following steps:

- Downloading, from the cartridge, the sample JCLs necessary to the execution of steps 2, 3, and 4. To do so, follow the instructions specified in the 'Program Directory for Pactables' document.
- Setting up the SMP/E environment and the 'Target zone' and 'Distribution zone' files of the HBVP350 root FMID:
  - 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 technical files DDDEF input (BVP5DDEF)
  - Defining the 'Target zone' and 'Distribution zone' files (BVP6DDEF)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP7ALLO)
- Implementing the 'Target zone' and 'Distribution zone' files of the JBVP353 dependent FMID:
  - Defining the 'Target zone' and 'Distribution zone' files (BVP8TDD)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP9TAL)
- Installing the components of the HBVP350 and JBVP353 FMIDs:
  - Execution of RECEIVE (BVPTREC)
  - Execution of APPLY (BVPTAPP)
  - Execution of ACCEPT (BVPTACC)

When the installation is executed in an environment where the HBVP350 FMID and a JBVP35x FMID are already installed, you just have to install the Pactables JBVP353 dependent FMID.

Then execute steps 3 and 4 once the necessary JCLs are downloaded as described in step 1 and modify the RECEIVE, APPLY and ACCEPT JCLs to remove the HBVP350 FMID from the concerned lines.

Once this phase has been executed, all the components required for the installation are to be found in the following PDSs ; 'hlq' indicates the common prefix of the elements supplied (High-Level Qualifier):

- hlq.SBVPINST: all the 80-long files, including:
  - The initial installation JCL (BVPTINIT)
  - A file which contains the installation and operation JCLs and procedures (BVPTTTAL)
  - The sample SMP/E JCLs
- hlq.SBVPTF2: TC initialization files
- hlq.SBVPMBR8: batch load-modules
- hlq.SBVPMTR8: on-line load-modules
- hlq.SBVPTF5: TE error messages file
- hlq.SBVPDBD: dbd
- hlq.SBVPPSB: psb

Phase 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 3,390 disk unit), 30 pads directory.

Phase 3

The third phase involves copying the 'hlq.SBVPINIT(BVPTINIT)' JCL into the PDS mentioned in phase 2, modifying its parameters to match the constraints of the site and executing it so as to obtain a complete installation and operating JCL.

The BVPTINIT JCL executes the BVPMMJCL program loaded in 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 phase.

- In the '//SYSUT2 DD DSN=' field, enter the name of the file in which the complete JCL is to be saved.

  This file can be either a PDS member initially created to save all the JCL's, 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 a further re-installation.

Initial JCL

```
//VAPACTAB  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(BVPTTTAL),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'
//*     -----------------------------------------------
//*   - CHANGE &HLQ BY THE NAME OF SMPE HLQ ON YOUR SYSTEM
//*
//*   - 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=PACT          .JOB NAMES PREFIXES (4 CHARACTERS MAX)
===PRM CCPT=<>            .JOB ACCOUNTING CODES (JOB CARDS)
===PRM CLASSJ=1           .JOB EXECUTION CLASS (JOB CARDS)
===PRM MSGCL=A            .JCL OUTPUT CLASS (MSGCLASS)
===PRM INDSV='EXP.BVP3V'   .SYSTEM VSAM FILES INDEX
===PRM INDSN='EXP.BVP3N'   .SYSTEM NON VSAM FILES INDEX
===PRM INDUV='UTI.BVP3V'   .USER VSAM FILES INDEX
===PRM INDUN='UTI.BVP3N'   .USER NON VSAM FILES INDEX
===PRM BASE='T350'        .CODE OF PACTABLES DATABASE   (4 CHAR)
===PRM OUT=H              .JOB SYSOUT CLASS
===PRM UWK=SYSDA          .WORK FILES UNIT
===PRM UNITSN=SYSDA       .NON VSAM SYSTEM FILES UNIT
===PRM UNITSV=SYSDA       .VSAM SYSTEM FILES UNIT
===PRM UNITUN=SYSDA       .NON VSAM USER FILES UNIT
```

```
===PRM UNITUV=SYSDA       .VSAM USER FILES UNIT
===PRM VOLSN=             .SYSTEM NON VSAM FILES VOL=SER=
===PRM VOLSV=             .SYSTEM VSAM FILES VOLUME
===PRM VOLUN=             .USER NON VSAM FILES VOL=SER
===PRM VOLUV=             .USER VSAM FILES VOLUME
===PRM VCAT=              .USER FILES VSAM CATALOG
===PRM SCAT=              .SYSTEM FILES VSAM CATALOG
===PRM BIBP='SYS1.PROCLIB'  .PROCEDURE LIBRARY
===PRM DSCB='BVP.DSCB'      .DSCB MODEL FILE
===PRM HLQ='&HLQ',          .HIGH LEVEL QUALIFIER OF LOAD MODULES
===PRM LNG='E'            .DATABASE LANGAGE (E=ENGLISH, F=FRENCH)
===PRM BIBT='SYS1.SORTLIB'     .SORT LIBRARY
===PRM BCOB='SYS1.SCEERUN'     .COBOL ROUTINE LIBRARY
===PRM DSMS=<>                 .INDEX OF PRODUCT ELEMENTS (DC)
===PRM TRANT='PT350'           .TRANSACTION CODE TO PACTABLES
===PRM TRANP='PT350W'          .TRANSACTION CODE TO PARAMETERS
===PRM REGSIZ='1536K'        .SIZE OF THE REGION FOR BATCH PROCED
===PRM IMSID='IM35'          .'IMSID' PARAMETER FOR BMP PROCEDURE
===PRM IRLM=N                .USING IRLM IN BATCH PROCEDURES (N=N
===PRM DBRC=N                .USING DBRC IN BATCH PROCEDURES (N=N
===PRM SUG='35'                .SUFFIX OF VA PAC'S PSB
===PRM ROOT='P3'              .ROOT OF THE SYSTEM (2 CHARACTERS)
===PRM HEXA='D7F3'             .ROOT OF THE SYSTEM IN HEXA
===PRM DBDLIB='&HLQ.SBVPDBD'   .DBD LIBRARY
===PRM PSBLIB='&HLQ.SBVPPSB'   .PSB LIBRARY
===PRM ACBLIB='&HLQ.ACBLIB'   .ACB LIBRARY
===PRM RESLIB='IMSVS.RESLIB'   .IMS RESLIB
===PRM PRCLIB='IMSVS.PROCLIB'  .IMS PROCLIB
===PRM CLS='2'                 .CLASS FOR TRANSACTION CODES
===BEGMOD
./ ADD NAME=$ZMODUL
/*
//
```

Complete JCL installation

The BVPMMJCL module reads the BVPTTTAL JCL skeleton file and outputs a complete JCL. It allows you to:

- Select the installation language F (French) or E (English),
- 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 after the JCL modules to separate them.

This step can be executed as many times as necessary to generate a complete JCL.

User input

Refer to the following paragraphs:

- Coding of BVPMMJCL commands
- Table of JCL modules
- Parameter chart
- JCL separators

Output result: Complete JCL

The resulting SYSUT2 file contains all the installation and operation JCLs. This file may be modified (if necessary) via a text editor before installation begins.

Two types of operations are to be performed on the complete JCL:

1. Global modifications (if necessary):

Adaptations can be performed on all the JCLs.

VSAM CATALOGS are entered as comments in the installation JCL:

- In DELETE/DEFINEs, as:

```
/*: CATALOG ($VCAT) */
/*: CATALOG ($SCAT) */
```

- In JCL STEPCATs as:

```
//*: STEPCAT  DD
//*:          DD
```

- In procedure parameters as:

```
//*: VSAMCAT='$VCAT'
//*: SYSTCAT='$SCAT'
```

When these parameters are not required on the site, the resulting JCL may remain unchanged.

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

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

Blocking criteria for large files can also be changed.

Caution: SMS

- In the installation jobs which include 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).

In most cases, it is recommended to perform general modifications on the JCLs before splitting these JCLs.

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 that is found (see the following table of JCL modules).

So the complete JCL can be split in as many members as JCL modules in a PDS. The complete JCL file is to be used as SYSIN for 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 done the other way round before executing jobs which contain IEBUPDTE.

Printed output

BVPMMJCL outputs a list for each JCL module created, with the parameters taken into account.

Note:

Since the JCL skeleton parameters 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.

Coding of BVPMMJCL commands

```
===SELM mmmm1 mmmm2 ...   .Selection of JCL modules
                           mmmm1 = name of JCL module
                           mmmm2 = name of JCL module
                           etc.
                           If there is no ===SELM line, all
                           the JCL modules are selected.
===PRM  PPPP=pppp         .Parameter
                           PPPP = name of parameter
                           pppp = value of parameter
```

Note: On ===PRM or ===SELV lines, comments may be entered. They must be preceded by a period and not exceed column 72.

```
===BEGMOD         Insertion of lines at beginning of module.
....1         )
.....         ) lines to insert before each module
....n          )
===ENDMOD         Insertion of lines at end of module.

....1         )
.....         ) lines to insert after each module
....n          )
```

Installation default settings

- PARAMETERS (===PRM):

The values indicated are examples; they should be replaced according to the site's specific needs.

- MODULES (===SELM):

No selection; all modules are copied.

- JCL MODULE FIRST LINE (===BEGMOD):

```
./ ADD NAME=$MODULE
```

This adds a line before each JCL module, in the form:

```
./ ADD NAME=<name-of-JCL-module>
```

JCL modules

Table of installation JCLs

Member	Contents	Procedure
	Installation of the System and of the Administration Database	
PACTSY	Allocation and loading of the parameters PDS	
PACTPROI	Proclib allocation (optional)	
PACTPROC	Proclib loading	
PACTDBD	Allocation and loading of the DBD sources PDS	
PACTPSB	Allocation and loading of the PSD sources PDS	
PACTSYS	Zap	
PACTTE	Loading of the error messages and online help	LDTE

Member	Contents	Procedure
PACTTZ	Initialization of the Spas database	LDTZ
PACTTB	Initialization of PUF work database	PUF
PACTTC	Loading of the test database	
PACTTAB	Restoration of the test database	RSTA
PACTACB	ACBs compilation	

Member	Contents	Proc	Nature
	Batch tests		
JCLGETT	Generation of tables	GETT	JCL OS
JCLINTA	Ex. of table initialization	INTA	JCL OS
JCLPRTA	Ex. of table printing	PRTA	JCL OS
JCLIMTA	Ex. of table import	IMTA	JCL OS
JCLUPTA	Ex. of table update	UPTA	JCL OS
JCLSVTA	Ex. of table saving	SVTA	JCL OS
JCLTCTA	Ex. of table migration	TCTA	JCL OS
JCLRSTA	Ex. of table restoration	RSTA	JCL OS
JCLRETA	Ex. of table reorganization	RETA	JCL OS
JCLPMTA	Ex. of parameter update	PMTA	JCL OS
JCLEXTA	Ex. of table extraction	EXTA	JCL OS
JCLTUTA	Ex. of direct consultation	TUTA	JCL OS
JCLBVTA	Ex. of database transfer onto files	BVTA	JCL OS
JCLLDRE	Ex. of file transfer onto databases	LDRE	JCL OS
JCLLDTA	Ex. of description lists printing	LDTA	JCL OS
JCLCDT1	Ex. of description comparison 1	CDT1	JCL OS
JCLCDT2	Ex. of description comparison 2	CDT2	JCL OS
JCLCVTA	Ex. of table contents update	CVTA	JCL OS
JCLLPTA	Ex. of programs list	LPTA	JCL OS

JCL parameterization

Syntax:

===PRM PPPP=pppp .Comment

- The parameter values which contain special characters must be bounded by single quotes.
- Comments on ===PRM lines should not exceed column 72. They should be preceded by a '.' (period).

Note :

When the default value or the '<>' value is entered, the paremeter is required.

Parameters table

Parameter	Meaning	Default
PRFJ	Prefix of job names (5 characters maxi)	PACT
CCPT	Job accounting class	<>
CLASSJ	Job execution class	1
MSGCL	JCL output class	A
INDSV	Pactables VSAM system (24 char. maxi)	'EXP.BVP3V'
INDSN	Pactables non-VSAM system (SAM, PDS) (24 char. maxi)	'EXP.BVP3N'
INDUV	VSAM user (24 char. maxi)	'UTI.BVP3V'
INDUN	Non-VSAM user (SAM) (24 char. maxi)	'UTI.BVP3N
BASE	Pactables database code	T350
OUT	SYSOUTs printing class	H
UWK	UNIT of work files used	SYSDA
UNITSN	UNIT of non-VSAM System files	SYSDA
UNITSV	UNIT of VSAM System files	SYSDA
UNITUN	UNIT of non-VSAM User files	SYSDA
UNITUV	UNIT of VSAM User files	SYSDA
VOLSN	VOL=SER Non-VSAM System files	
VOLSV	VOL=SER VSAM System files	
VOLUN	VOL=SER non-VSAM User files	
VOLUV	VOL=SER VSAM User files	
VCAT	VSAM catalog of development database (user files)	
SCAT	System VSAM catalog (system files)	
DSCB	DSNAME of DSCB model file for generation files	'BVP.DSCB'
BIBP	DSNAME of procedures library	'SYS1.PROCLIB'

Parameter	Meaning	Default
HLQ	Prefix of batch and TP load-modules library (30 char. maximum)	'HLQ'
LNG	Pactable database language code (E = English, F = French)	'E'
BIBT	DSNAME of sort library (SORTLIB)	'SYS1.SORTLIB'
BCOB	DSNAME of Pactables COBOL routines	'SYS1.SCEERUN'
	library	
TRANT	Pactables transaction code	PT350
TRANP	Pactables Administration transaction code	PW350
REGSIZ	Region size for batch procedures	1536K
IMSID	IMSID parameter for batch procedures (N=NO)	N
IRLM	ILRM use in batch procedures (N=NO)	N
DBRC	DBRC use in batch procedures (N=NO)	N
SUG	Suffix of PSBs and DBDs (2 characters)	35
ROOT	System root (2 characters)	P3
HEXA	System root in hexadecimal	D7F3
DBDLIB	VA Pac DBDs library (2)	$HLQ.DBDLIB
PSBLIB	VA Pac PSBs library (2)	$HLQ.PSBLIB
ACBLIB	ACBs library	EXP.ACBLIB
RESLIB	IMS RESLIB	IMSVS.RESLIB
PRCLIB	IMS PROCLIB	IMSVS.PROCLIB
CLS	Class for transaction codes	2

(1) These file DSNs should be replaced with those already installed on the site only if the Pactables or DSMS modules are installed or if the default name is not appropriate.

(2) Warning: if the suffix chosen ($SUG parameter) is equal to 35, then the prefix of the $DBDLIB and $PSBLIB parameters must be equal to the value of the $HLQ parameter followed by SBVPDBD for the $DBDLIB parameter and SBVPPSB for the $PSBLIB parameter.

Example: if the $HLQ parameter has the value EXP.PT350, the value of the $DBDLIB parameter will have to be EXP.PT350.SBVPDBD and that of the $PSBLIB parameter will have to be EXP.PT350.SBVPPSB.

JCL module separators

```
===BEGMOD
....1    )
.....    ) lines to insert before each JCL module
....n    )
===ENDMOD
....1    )
.....    ) lines to insert after each JCL module
....n    )
```

Lines may be inserted as input in the BVPMMJCL if the default option is not appropriate (see Subchapter ' 'Installation Default settings' above).

The purpose of these lines is to execute the separation of the JCL file created by the BVPMMJCL utility into as many members as there are JCL modules.

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

Installation process

Once all the JCLs are obtained, you can install Pactables:

- Allocate and load the parameters PDS,
- Allocate the procedures library,
- Catalog the operations procedures,
- Catalog the DBDs sources,
- Catalog the PSBs sources,
- Zap,
- Initialize and load the file which contains the error messages and online help,
- Initialize the Spas database,
- Initialize the TUF work database,
- Initialize and load the test database,
- Restore the test database,
- Generate the ACBs.

Allocation and loading of system parameters

PACTSY module: '$prfj.SY' job

Step	Program	Comment
STEP1	IDCAMS	Delete of the parameters PDS
STEP2	IEFBR14	Reservation of the SY parameters PDS

Step	Program	Comment
STEP3	IEBUPDTE	Loading of the PDS which contains the input for the IDCAMS

The library contains the following members:

Member	Contents or format
DFBVPTE	DELETE/DEFINE TE
DFBVPTZ	DELETE/DEFINE TZ
DF&BASE.AD	DELETE/DEFINE AD
DF&BASE.AV	DELETE/DEFINE AV
DF&BASE.TD	DELETE/DEFINE TD
DF&BASE.TU	DELETE/DEFINE TU
DF&BASE.TV	DELETE/DEFINE TV
DF&BASE.TG	DELETE/DEFINE TG
DF&BASE.TW	DELETE/DEFINE TW
DF&BASE.TS	DELETE/DEFINE TS
DF&BASE.TB	DELETE/DEFINE TB
DF&BASE.T1	DELETE/DEFINE T1
VERIFTD	VERIFY (PAC7TD)
VERIFTE	VERIFY (PAC7TE)
VERIFTG	VERIFY (PAC7TG)
VERIFTS	VERIFY (PAC7TS)
VERIFTU	VERIFY (PAC7TU)
VERIFTV	VERIFY (PAC7TV)
VERIFTW	VERIFY (PAC7TW)
LDBVPTZ	TZ maxi. record
LD&BASE.TB	TB maxi. record
LD&BASE.TG	TG maxi. record
SY&BASE.TG	
MAXKEY	maxi. record
REPROTB	TB IDCAMS input
DFSVSAM8	
DFSVSAM9	
PACTTRL	to be declared on IMS for TP mode

Note: This PDS contains the definitions of the database files. These sizes are allowed for the installation test database. They must be adapted by the user according to the evolution of the database size.

Execution JCL

```
//$PRFJ.SY JOB ($CCPT),'PAC PACTSY',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//* INSTALLATION  : LOADING PARAMETERS IN PDS                         *
//*                                                                   *
//*-------------------------------------------------------------------*
//*    *               I M P O R T A N T                 *          *
//*    *                  -----------                    *          *
//*    *      BEFORE EXECUTING THIS JOB, REPLACE ALL     *          *
//*    *          ':/' WITH './' VIA THE EDITOR.         *          *
//*    *-------------------------------------------------*          *
//*                                                                   *
//*-------------------------------------------------------------------*
//*    *                   N O T E                       *          *
//*    *                   -------                       *          *
//*    * THESE PARAMETERS CONTAIN THE SYSINS FOR ALLOCA-  *          *
//*    * TING FILES AND DATABASES USED IN THE PACTABLES   *          *
//*    * MANAGEMENT FUNCTION. THE SIZES INDICATED CAN BE  *          *
//*    * ADAPTED TO YOUR REQUIREMENTS.                    *          *
//*    *-------------------------------------------------*          *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDUN..BVPSY)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDUN..BVPSY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(6080,(100,,10))
//*
//STEP3   EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..BVPSY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/ ADD NAME=DF$BASE.AD
 DELETE ($INDUV..$BASE.AD) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.AD)       -
                  SHR (2,3)  RUS KEY (21,0)           -
                  VOL ($VOLUV)  CYL (2,1)              -
                  RECSZ (240,240) )                   -
        INDEX   ( NAME ($INDUV..$BASE.AD.I)   -
                  CISZ (1024) )                       -
        DATA    ( NAME ($INDUV..$BASE.AD.D)   -
                  FSPC (10,5)                         -
                  CISZ (2048) ) /*: CATALOG ($VCAT) :*/
```

```
:/ ADD NAME=DF$BASE.AV
 DELETE ($INDUV..$BASE.AV) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.AV)       -
                  SHR (2,3)  RUS KEY (35,4)           -
                  VOL ($VOLUV)  CYL (2,1)              -
                  RECSZ (80,1100) )                   -
        INDEX   ( NAME ($INDUV..$BASE.AV.I)   -
                  CISZ (1024) )                       -
        DATA    ( NAME ($INDUV..$BASE.AV.D)   -
                  FSPC (10,5)                         -
                  CISZ (2048) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DF$BASE.DD
 DELETE ($INDUV..$BASE.DD) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.DD)       -
                  SHR (2,3)  RUS KEY (21,0)           -
                  VOL ($VOLUV)  CYL (2,1)              -
                  RECSZ (240,240) )                   -
        INDEX   ( NAME ($INDUV..$BASE.DD.I)   -
                  CISZ (1024) )                       -
        DATA    ( NAME ($INDUV..$BASE.DD.D)   -
                  FSPC (10,5)                         -
                  CISZ (2048) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DF$BASE.TD
 DELETE ($INDUV..$BASE.TD) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TD)     -
                  SHR (2,3)  RUS  KEY (21,6)        -
                  VOL ($VOLUV)  CYL (2,1)            -
                  RECSZ (248,248) )                 -
        INDEX   ( NAME ($INDUV..$BASE.TD.I) -
                  CISZ (4096) )                     -
        DATA    ( NAME ($INDUV..$BASE.TD.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DF$BASE.TS
 DELETE ($INDUV..$BASE.TS) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TS)     -
                  SHR (2,3)  RUS  KEY (21,0)        -
                  VOL ($VOLUV)  CYL (2,1)            -
                  RECSZ (240,240) )                 -
        INDEX   ( NAME ($INDUV..$BASE.TS.I) ) -
        DATA    ( NAME ($INDUV..$BASE.TS.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DFBVPTE
 DELETE ($INDSV..BVPTE) CL
 DEFINE CLUSTER ( NAME ($INDSV..BVPTE)     -
                  SHR (2,3)  RUS  KEYS (17,6)       -
                  VOL ($VOLSV)  CYL (2,1)            -
                  RECSZ (098,098) )                 -
        INDEX   ( NAME ($INDUV..BVPTE.I) -
                  CISZ (4096) )                     -
        DATA    ( NAME ($INDUV..BVPTE.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($CATV) :*/
:/ ADD NAME=DF$BASE.TG
```

```
 DELETE ($INDUV..$BASE.TG) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TG)     -
                  SHR (2,3)  RUS  KEYS (22,6)       -
                  VOL ($VOLUV)  CYL (2,1)            -
                  RECSZ (094,094) )                 -
        INDEX   ( NAME ($INDUV..$BASE.TG.I) -
                  CISZ (4096) )                     -
        DATA    ( NAME ($INDUV..$BASE.TG.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DF$BASE.TU
 DELETE ($INDUV..$BASE.TU) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TU)     -
                  SHR (2,3)  RUS  KEYS (35 5)       -
                  INDEXED                           -
                  VOL ($VOLUV)     CYL (1,1)         -
                  RECSZ (040 040) )                 -
        INDEX   ( NAME ($INDUV..$BASE.TU.I) -
                  CISZ (1024) )                     -
        DATA    ( NAME ($INDUV..$BASE.TU.D) -
                  CISZ (1024) ) /*: CATALOG ($VCAT) :*/

:/ ADD NAME=DF$BASE.TV
 DELETE ($INDUV..$BASE.TV) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TV)     -
                  SHR (2,3)  RUS                    -
                  NONINDEXED                        -
                  VOL ($VOLUV) CYL (2,1)             -
                  RECSZ (4089 4089) )               -
        DATA    ( NAME ($INDUV..$BASE.TV.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DF$BASE.TW
 DELETE ($INDUV..$BASE.TW) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TW)     -
                  SHR (2,3)  RUS  KEY (35,4)        -
                  VOL ($VOLUV)  CYL (2,1)            -
                  RECSZ (80,1100) )                 -
        INDEX   ( NAME ($INDUV..$BASE.TW.I) ) -
        DATA    ( NAME ($INDUV..$BASE.TW.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DFBVPTZ
 DELETE ($INDSV..BVPTZ) CL
 DEFINE CLUSTER ( NAME ($INDSV..BVPTZ)       -
                  SHR (2,3)    KEYS (11,6)           -
                  VOL ($VOLSV)  CYL (3 3)             -
                  RECSZ (8510 8510) RUS )           -
        INDEX   ( NAME ($INDSV..BVPTZ.I)   -
                  CISZ (4096) )                     -
        DATA    ( NAME ($INDSV..BVPTZ.D)   -
                  FSPC (10,5)                       -
                  CISZ (10240) ) /*: CATALOG ($CATV) :*/
:/ ADD NAME=DF$BASE.T1
 DELETE ($INDUV..$BASE.T1) CL
```

```
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.T1)     -
                  SHR (2,3)  RUS  KEYS (63 5)       -
                  INDEXED                           -
                  VOL ($VOLUV)     CYL (1,1)         -
                  RECSZ (068 068) )                 -
        INDEX   ( NAME ($INDUV..$BASE.T1.I) -
                  CISZ (1024) )                     -
        DATA    ( NAME ($INDUV..$BASE.T1.D) -
                  CISZ (1024) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=DF$BASE.TB
 DELETE ($INDUV..$BASE.TB) CL
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.TB)     -
                  SHR (2,3)  RUS                    -
                  NONINDEXED                        -
                  VOL ($VOLUV) CYL (2,1)             -
                  RECSZ (4089 4089) )               -
        DATA    ( NAME ($INDUV..$BASE.TB.D) -
                  FSPC (10,5)                       -
                  CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/ ADD NAME=VERIFTD
 VERIFY FILE (PAC7TD)
:/ ADD NAME=VERIFTS
 VERIFY FILE (PAC7TS)
:/ ADD NAME=VERIFTE
 VERIFY FILE (PAC7TE)
:/ ADD NAME=VERIFTG
 VERIFY FILE (PAC7TG)
:/ ADD NAME=VERIFTU
 VERIFY FILE (PAC7TU)
:/ ADD NAME=VERIFTV
 VERIFY FILE (PAC7TV)
:/ ADD NAME=VERIFTW
 VERIFY FILE (PAC7TW)
:/ ADD NAME=REPROTB
 REPRO INFILE(INTB) OUTFILE(OUTTB)
:/ ADD NAME=BL$BASE.TC
 DEFINE GENERATIONDATAGROUP -
        (NAME ($INDUVQ..$BASE.TC) LIMIT (3) SCR)
:/ ADD NAME=LD$BASE.TG
999999999999999999999 RECORD OF LOADING TG DATABASE
:/ ADD NAME=LDBVPTZ
99999999ZZ9 RECORD OF LOADING DATABASE SPA FOR PACTABLES
:/ ADD NAME=LD$BASE.TB
99999999999999999999999999999999999999999999999999999999999999999999
:/ ADD NAME=SY$BASE.TG
   GENERATE    MAXFLDS=2,MAXLITS=5
   RECORD      FIELD=(80,1,,1),
               FIELD=(5,'     ',,81)
:/ ADD NAME=DFSVSAM8
4096,8
:/ ADD NAME=DFSVSAM9
12288,9
:/ ADD NAME=PACTCTRL
*********************************************************************
*           DECLARATION OF DBD                                      *
```

```
*****************************************************************
 DATABASE ACCESS=UP,DBD=(PACDTD$SUG,PACDTE$SUG,PACDTG$SUG)
 DATABASE ACCESS=UP,DBD=(PACDTU$SUG,PACDTV$SUG,PACDTZ$SUG)
 DATABASE ACCESS=UP,DBD=(PACDT1$SUG,PACDTB$SUG)
*****************************************************************
*           DECLARATION OF TRANSACTIONS                         *
*****************************************************************
 APPLCTN  PSB=BVPPLNK
 TRANSACT CODE=$ROOT.CLNK,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP501
 TRANSACT CODE=$TRANT,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,     $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP601
 TRANSACT CODE=$TRANP,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,     $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP500
 TRANSACT CODE=$ROOT.C500,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP510
 TRANSACT CODE=$ROOT.C510,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP520
 TRANSACT CODE=$ROOT.C520,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP530
 TRANSACT CODE=$ROOT.C530,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP540
 TRANSACT CODE=$ROOT.C540,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP550
 TRANSACT CODE=$ROOT.C550,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP560
 TRANSACT CODE=$ROOT.C560,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP570
 TRANSACT CODE=$ROOT.C570,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP580
 TRANSACT CODE=$ROOT.C580,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
```

```
 APPLCTN  PSB=BVPP590
 TRANSACT CODE=$ROOT.C590,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP599
 TRANSACT CODE=$ROOT.C599,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP600
 TRANSACT CODE=$ROOT.C600,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP610
 TRANSACT CODE=$ROOT.C610,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
 APPLCTN  PSB=BVPP620
 TRANSACT CODE=$ROOT.C620,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $Y
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
          MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
*****************************************************************
*           DECLARATION OF BMP                                  *
*****************************************************************
 APPLCTN  PSB=PTA250$SUG,PGMTYPE=BATCH
 APPLCTN  PSB=PTA300$SUG,PGMTYPE=BATCH
PP
//
```

Loading of the procedures

It is recommended that all operation procedures be cataloged in one procedures 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.

PACTPROI module: '$prfj.PROI' job

Allocation of a reserved library (optional)

Step	Program	Comments
STEP1	IEFBR14	Allocation of procedures library

Execution JCL

```
//$PRFJ.PROI  JOB ($CCPT),'PAC PACTPROI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*              INSTALLATION - PACTPROI
```

```
//*                                                                   *
//*                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=$BIBP,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(200,20,10))
```

PACTPRO module: '$prfj.PRO' job

Loading of procedures

This job includes the 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.

Procedures are detailed in the other chapters of this manual.

Execution JCL

```
//$PRFJ.PROC JOB ($CCPT),'PROCEDURES ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*        --- CATALOGING OF VA PACTABLES PROCEDURES ---              *
//*                                                                   *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
```

```
//  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2 DD DSN=$BIBP,DISP=SHR
//SYSIN   DD   DATA,DLM='F+'
:/ ADD NAME=BVPCDT1
:/ ADD NAME=BVPCDT2
:/ ADD NAME=BVPCVTA
:/ ADD NAME=BVPEXTA
:/ ADD NAME=BVPGETA
:/ ADD NAME=BVPGETD
:/ ADD NAME=BVPGETI
:/ ADD NAME=BVPGETT
:/ ADD NAME=BVPIMTA
:/ ADD NAME=BVPINTA
:/ ADD NAME=BVPLDTA
:/ ADD NAME=BVPLPTA
:/ ADD NAME=BVPPMTA
:/ ADD NAME=BVPPRTA
:/ ADD NAME=BVPRETA
:/ ADD NAME=BVPRSTA
:/ ADD NAME=BVPSMTD
:/ ADD NAME=BVPSVTA
:/ ADD NAME=BVPTCTA
:/ ADD NAME=BVPTUTA
:/ ADD NAME=BVPUPTA
F+
//
```

Copy of access sub-programs

PACTLOAD module: '$prfj.LOAD' job

Copy of the BVPP820 and BVPP920 load-modules from the batch load-modules library to the on-line load-modules library.

These tables access programs are used in user checks.

Execution JCL

```
//$PRFJ.LOAD JOB ($CCPT),'LOAD',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*               INSTALLATION - PACTLOAD
//*       - COPY BVPP920 AND BVPP820 IN BATCH LIBRARY
//*****************************************************************
//STEP1    EXEC PGM=IEBCOPY,REGION=0M
//SYSPRINT DD SYSOUT=$OUT
//I1       DD DSN=$HLQ..SBVPMTR8,DISP=SHR
//O1       DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//SYSIN    DD *
   C INDD=I1,OUTDD=O1
```

```
     S M=BVPP820
     S M=BVPP920
/*
//*
```

Loading of the DBD sources

PACTDBD module: '$prfj.SRD' job

This step is to be executed if, upon the installation, the default value of the $SUG parameter has not been kept.

The default value of the $SUG parameter is: 35.

In this case, all the sources must be compiled again.

The '$PRFJ.SRCD' job allocates the $INDUN..$BASE.SRCD library and updates, via IEBUPDTE, the DBD sources in this library.

The DBD sources are cataloged under the name: PACDTxxyy

with xx = database code and y = value of the $SUG parameter.

Execution JCL

```
//$PRFJ.SRD JOB ($CCPT),'PDS-SRC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                                                                   *
//* INSTALLATION  : LOADING DBD'S SOURCES                             *
//*                                                                   *
//*                                                                   *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*                                                                   *
//*********************************************************************
//*
//STEP1   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
    DELETE ($INDUN..$BASE.SRCD)
//*
//STEP2   EXEC PGM=IEFBR14
//DDA      DD DSN=$INDUN..$BASE.SRCD,DISP=(,CATLG,DELETE),
//            UNIT=$UNITUN,VOL=SER=$VOLUN,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(250,20,20))
//*
//STEP3  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDUN..$BASE.SRCD,DISP=SHR
//SYSIN    DD DATA,DLM='F+'
```

```
:/ ADD NAME=PACDTD$SUG
       DBD     NAME=PACDTD$SUG,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TD$SUG,DEVICE=3390,                        $Y
               RECORD=248,SIZE=4096
       SEGM    NAME=PAC7TD,BYTES=240
       FIELD   NAME=(CLETD,SEQ,U),BYTES=21,START=1
       DBDGEN
       END
:/ ADD NAME=PACDTE$SUG
       DBD     NAME=PACDTE$SUG,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TE$SUG,DEVICE=3390,                        $Y
               RECORD=98,SIZE=4096
       SEGM    NAME=PAC7TE,BYTES=90
       FIELD   NAME=(CLETE,SEQ,U),BYTES=17,START=1
       DBDGEN
       END
:/ ADD NAME=PACDTG$SUG
       DBD     NAME=PACDTG$SUG,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TG$SUG,DEVICE=3390,                        $Y
               RECORD=94,SIZE=4096
       SEGM    NAME=PAC7TG,BYTES=85
       FIELD   NAME=(CLETG,SEQ,U),BYTES=22,START=1
       DBDGEN
       END
:/ ADD NAME=PACDTU$SUG
       DBD     NAME=PACDTU$SUG,ACCESS=(INDEX,VSAM)
       DATASET DD1=PAC7TU$SUG,DEVICE=3390
       SEGM    NAME=IPAC7TU,BYTES=35
       FIELD   NAME=(CLETU,SEQ,U),BYTES=35,START=1
       LCHILD  NAME=(PAC7TV,PACDTV$SUG),INDEX=CLETV
       DBDGEN
       END
:/ ADD NAME=PACDTV$SUG
       DBD     NAME=PACDTV$SUG,ACCESS=(HIDAM,VSAM)
       DATASET DD1=PAC7TV$SUG,DEVICE=3390,                        $Y
               SIZE=4096
       SEGM    NAME=PAC7TV,BYTES=(1100,0080)
       FIELD   NAME=(CLETV,SEQ,U),BYTES=35,START=5
       LCHILD  NAME=(IPAC7TU,PACDTU$SUG),PTR=INDX
       DBDGEN
       END
:/ ADD NAME=PACDTZ$SUG
       DBD     NAME=PACDTZ$SUG,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TZ$SUG,DEVICE=3390,                        $Y
               RECORD=8510,SIZE=10240
       SEGM    NAME=PAC7TZ,BYTES=8500
       FIELD   NAME=(CLETZ,SEQ,U),BYTES=11,START=1
       DBDGEN
       END
:/ ADD NAME=PACDT1$SUG
       DBD     NAME=PACDT1$SUG,ACCESS=(INDEX,VSAM)
       DATASET DD1=PAC7T1$SUG,DEVICE=3390
       SEGM    NAME=IPAC7T1,BYTES=63
       FIELD   NAME=(CLET1,SEQ,U),BYTES=63,START=1
       LCHILD  NAME=(PAC7TB,PACDTB$SUG),INDEX=CLETB
```

```
       DBDGEN
       END
:/ ADD NAME=PACDTB$SUG
       DBD     NAME=PACDTB$SUG,ACCESS=(HIDAM,VSAM)
       DATASET DD1=PAC7TB$SUG,DEVICE=3390,                    $Y
               SIZE=4096
       SEGM    NAME=PAC7TB,BYTES=(1140,0080)
       FIELD   NAME=(CLETB,SEQ,U),BYTES=63,START=4
       LCHILD  NAME=(IPAC7T1,PACDT1$SUG),PTR=INDX
       DBDGEN
       END
F+
//
```

Loading of the PSB sources

PACTPSB module: 'prfj.SRP' job

This step is to be executed if, upon the installation, the default values of the $ROOT and $SUG parameters have not been kept.

The default values of these parameters are:

- 35 for the $SUG parameter
- P3 for the $ROOT parameter

  In this case, all the sources must be compiled again.

  The '$PRFJ.SRCP' job allocates the $INDUN..$BASE.SRCP library and updates, via IEBUPDTDTE the PSB sources in this library. The PSB sources are cataloged under the name: xxxxxyy

  with xxxxxx = VA Pac program code and y = value of the $SUG parameter.

  These are the Batch PSBs.

- BVPyyyy , where yyyy can have the following values:

  P500 - P501 - P510 - P520 - P530 - P540 - P550 - P560 - P570 - P580 - P590 - P599 -P600 - P601 - P610 - P620 - PLNK.

  These are the online PSBx;

Execution JCL

```
//$PRFJ.SRP JOB ($CCPT),'PDS-SRC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*                                                                   *
//* INSTALLATION  : LOADING PSB'S SOURCES                             *
//*                                                                   *
//*                                                                   *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*                                                                   *
//*********************************************************************
//*
```

```
//STEP1   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
    DELETE ($INDUN..$BASE.SRCP)
//*
//STEP2   EXEC PGM=IEFBR14
//DDA      DD DSN=$INDUN..$BASE.SRCP,DISP=(,CATLG,DELETE),
//            UNIT=$UNITUN,VOL=SER=$VOLUN,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(20,20,20))
//*
//STEP3  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDUN..$BASE.SRCP,DISP=SHR
//SYSIN    DD DATA,DLM='F+'
:/ ADD NAME=PTABVS$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTABVS$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAD05$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTAD05$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAD10$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PSBGEN  PSBNAME=PTAD10$SUG,LANG=COBOL,CMPAT=YES
       END

:/ ADD NAME=PTAD20$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=A,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=A,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTAD20$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAINI$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=L,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=L,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTAINI$SUG,LANG=COBOL,CMPAT=YES
       END
```

```
:/ ADD NAME=PTAREO$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=L,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=L,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTAREO$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTARSG$SUG
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=L,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTARSG$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTARSV$SUG
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=L,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTARSV$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTARST$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=L,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PSBGEN  PSBNAME=PTARST$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTASVG$SUG
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GE,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTASVG$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTASVV$SUG
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GE,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTASVV$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTASVT$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GE,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PSBGEN  PSBNAME=PTASVT$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAT10$SUG
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=L,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PSBGEN  PSBNAME=PTAT10$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAT15$SUG
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=L,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTAT15$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAT17$SUG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=L,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=PTAT17$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAT19$SUG
       PCB     TYPE=DB,DBDNAME=PACDTB$SUG,PROCOPT=L,KEYLEN=63
       SENSEG    NAME=PAC7TB
```

```
       PSBGEN  PSBNAME=PTAT19$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAU80$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG  NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG  NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
       SENSEG  NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
       SENSEG  NAME=PAC7TG
       PSBGEN  PSBNAME=PTAU80$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAV10$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG  NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG  NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
       SENSEG  NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
       SENSEG  NAME=PAC7TG
       PSBGEN  PSBNAME=PTAV10$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTAV20$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG  NAME=PAC7TD
       PSBGEN  PSBNAME=PTAV20$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PACXDT$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG  NAME=PAC7TD
       PSBGEN  PSBNAME=PACXDT$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTA100$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=A,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTA100$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTA120$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTA120$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTA150$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG    NAME=PAC7TE
```

```
        PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
        SENSEG    NAME=PAC7TG
        PSBGEN  PSBNAME=PTA150$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PTA160$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PSBGEN  PSBNAME=PTA160$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PTA250$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=A,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=A,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PSBGEN  PSBNAME=PTA250$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PTA290$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PSBGEN  PSBNAME=PTA290$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PTA300$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=A,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
        SENSEG    NAME=PAC7TG
        PSBGEN  PSBNAME=PTA300$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PTA310$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=A,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
        SENSEG    NAME=PAC7TG
        PSBGEN  PSBNAME=PTA310$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PTA320$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
        SENSEG    NAME=PAC7TG
        PSBGEN  PSBNAME=PTA320$SUG,LANG=COBOL,CMPAT=YES
```

```
       END
:/ ADD NAME=PTA350$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PSBGEN  PSBNAME=PTA350$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTA360$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PSBGEN  PSBNAME=PTA360$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTA400$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=GOT,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=GOT,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=GOT,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PSBGEN  PSBNAME=PTA400$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=PTA420$SUG
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=GOT,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PSBGEN  PSBNAME=PTA420$SUG,LANG=COBOL,CMPAT=YES
       END
:/ ADD NAME=BVPPLNK
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=G,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPPLNK,LANG=COBOL
       END
:/ ADD NAME=BVPP500
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=A,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=G,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=A,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP500,LANG=COBOL
       END
:/ ADD NAME=BVPP501
       PCB  TYPE=TP,MODIFY=YES
```

```
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP501,LANG=COBOL
       END
:/ ADD NAME=BVPP510
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=A,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=G,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP510,LANG=COBOL
       END
:/ ADD NAME=BVPP520
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=A,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=G,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP520,LANG=COBOL
       END
:/ ADD NAME=BVPP530
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=G,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP530,LANG=COBOL
       END
:/ ADD NAME=BVPP540
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP540,LANG=COBOL
       END
:/ ADD NAME=BVPP550
```

```
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP550,LANG=COBOL
       END
:/ ADD NAME=BVPP560
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=G,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP560,LANG=COBOL
       END
:/ ADD NAME=BVPP570
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP570,LANG=COBOL
       END
:/ ADD NAME=BVPP580
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTV$SUG,PROCOPT=G,KEYLEN=35
       SENSEG    NAME=PAC7TV
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP580,LANG=COBOL
       END
:/ ADD NAME=BVPP590
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=A,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP590,LANG=COBOL
       END
:/ ADD NAME=BVPP599
       PCB  TYPE=TP,MODIFY=YES
```

```
       PSBGEN  PSBNAME=BVPP599,LANG=COBOL
       END
:/ ADD NAME=BVPP600
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=A,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=A,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP600,LANG=COBOL
       END
:/ ADD NAME=BVPP601
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP601,LANG=COBOL
       END
:/ ADD NAME=BVPP610
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=A,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP610,LANG=COBOL
       END
:/ ADD NAME=BVPP620
       PCB  TYPE=TP,MODIFY=YES
       PCB     TYPE=DB,DBDNAME=PACDTD$SUG,PROCOPT=G,KEYLEN=21
       SENSEG    NAME=PAC7TD
       PCB     TYPE=DB,DBDNAME=PACDTE$SUG,PROCOPT=G,KEYLEN=17
       SENSEG    NAME=PAC7TE
       PCB     TYPE=DB,DBDNAME=PACDTG$SUG,PROCOPT=A,KEYLEN=22
       SENSEG    NAME=PAC7TG
       PCB     TYPE=DB,DBDNAME=PACDTZ$SUG,PROCOPT=A,KEYLEN=11
       SENSEG    NAME=PAC7TZ
       PSBGEN  PSBNAME=BVPP620,LANG=COBOL
       END
F+
//
```

Zap

PACTSYS module: ′$prfj.SYS′ job

This step is to be executed only if you want to change the default value of the root (parameter $ROOT=P3). The new value must be coded in hexadecimal for the zaps of the $ROOT.P501 and $ROOT.P601 modules.

Execution JCL

```
//$PRFJ.TSYS JOB ($CCPT),'SYSTEM FILES LOAD',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*************************************************************
//*
//*-------------------------------------------------------------*
//*                    I M P O R T A N T                        *
//*                    -----------------                        *
//*   IT IS NOT NECESSARY TO EXECUTE THESE STEPS IF THE    '    *
//*      ON-LINE MODULES' PREFIX CONTAINS THE VALUE '35'        *
//*      (CHECK THE SUF PARAMETER).                             *
//*   IF THE PREFIX IS MODIFIED, THESE STEPS MUST BE EXECUTED.*
//*      TO DO THIS, THE 'HEXA' PARAMETER MUST BE CODED WITH    *
//*      THE CORRESPONDING HEXADECIMAL VALUE OF THE NEW         *
//*      PREFIX.                                                *
//*      FOR INST.:  IF THE NEW PREFIX IS: ROOT='AB', THE       *
//*             'HEXA' PARAMETER MUST BE CODED: HEXA='C1C2'     *
//*                                                            *
//*-------------------------------------------------------------*
//STEP1   EXEC PGM=IMASPZAP
//SYSPRINT DD SYSOUT=$OUT
//SYSLIB   DD DSN=$HLQ..SBVPMTR8,DISP=SHR
 NAME BVPP501 BVPP501
 VER 0397 D7F3
 REP 0397 $HEXA
//*
//STEP2   EXEC PGM=IMASPZAP
//SYSPRINT DD SYSOUT=$OUT
//SYSLIB   DD DSN=$HLQ..SBVPMTR8,DISP=SHR
 NAME BVPP601 BVPP601
 VER 0397 D7F3
 REP 0397 $HEXA
//*
//
```

Loading of error messages

PACTTE module: '$prfj.TE' job

Loading of TE, error messages database

Step	Program	Comment
STEP1	IEHPROGM	Scracth/Uncatlg of IN file, sequential image of the error messages
STEP2	IDCAMS	Copy of the error messsage file supplied to the IN file
STEP3	LDTE	LDTE procedure: loading of the error messages database

Execution JCL

```
//$PRFJ.TTE  JOB ($CCPT),'LOADING TE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//  JCLLIB ORDER=($BIBP)
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*********************************************************************
//*-------------------------------------------------------------
//*      DELETE OF THE 'TE' FILE SEQUENTIAL IMAGE
//*-------------------------------------------------------------
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..BVPIN)
//*
//*-------------------------------------------------------------
//*      LOADING OF THE 'TE' FILE SEQUENTIAL IMAGE   IMAGE
//*-------------------------------------------------------------
//STEP2   EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//TEI      DD DSN=$HLQ..SBVPTF5(BVPTE),DISP=SHR
//TEO      DD DSN=$INDSN..BVPIN,DISP=(,CATLG,DELETE),
//            VOL=SER=$VOLSN,UNIT=$UNITSN,
//            SPACE=(TRK,(20,10),RLSE),
//            DCB=(RECFM=FB,LRECL=90,BLKSIZE=6030)
//SYSIN    DD *
  REPRO INFILE (TEI) OUTFILE (TEO)
/*
//*-------------------------------------------------------------
//*      LOADING OF THE 'TE' DATABASE
//*-------------------------------------------------------------
//STEP3   EXEC BVPLDTE
//*
//
```

Initialization of the Spas database

PACTTZ module: '$prfj.TZ' job

Loading of TZ, Spas database.

This job is to be executed upon the first installation only.

Step	Procedure	Comment
STEP1	LDTZ	Loading of the Spas database

Execution JCL

```
//$PRFJ.TTZ  JOB ($CCPT),'LOADING TZ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//  JCLLIB ORDER=($BIBP)
//*****************************************************************
//* INITIAL LOADING OF SPA DATABASE 'TZ'
```

```
//* ----------------------------------
//* JOB TO BE RUN ONLY WHEN FIRST INSTALLING THE PACTABLES 3.5
//* FUNCTION.
//*****************************************************************
//STEP1   EXEC BVPLDTZ
//*
```

Initialization of the TUF work database

PACTTB module: '$prfj.TB' job

Loading of TB, TUF-TP work database.

This job is to be executed upon the first installation only.

Step	Procedure	Comment
STEP1	LDTB	Loading of the TUF-TP database

Execution JCL

```
//$PRFJ.TTB  JOB ($CCPT),'LOADING TB',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//  JCLLIB ORDER=($BIBP)
//*****************************************************************
//* INITIAL LOADING OF TUF-TP DATABASE 'TB'
//* ---------------------------------------
//* JOB TO BE RUN ONLY WHEN FIRST INSTALLING THE PACTABLES 3.5
//* FUNCTION.
//*****************************************************************
//STEP1   EXEC BVPLDTB
//*
```

Initialization of the test database

PACTTC module: '$prfj.TC' job

Step	Program	Comment
STEP1	IEHPROGM	SCRATCH UNCATLG of DSCB model file
STEP2	IEFBR14	Allocation of DSCB model file
STEP3	IDCAMS	GDG of TC file
STEP4	IEBGENER	Loading of test backup

Execution JCL

```
//$PRFJ.TTC JOB ($CCPT),'PREPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
//*   JOB TO BE RUN ONLY AT THE FIRST INSTALLATION OF THE PACTABL
//*   SYSTEM
//*    . BUILDING OF DSCB MODEL AND INDEX DATA-GROUP FOR BACKUP
```

```
//*    . LOADING OF TEST BACKUP ON 'TC' FILE
//*
//*  ->NOTE
//*    ----
//*    IF "SMS" IS INSTALLED, DELETE //GDGMOD DD STATEMENTS
//*****************************************************************
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($DSCB)
//*
//STEP2    EXEC PGM=IEFBR14
//DSCB     DD DISP=(,CATLG),SPACE=(TRK,0),
//            UNIT=$UNITSN,
//            VOL=SER=$VOLSN,
//            DSN=$DSCB
//*
//STEP3    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDUN..$BASE.TC,
//         DISP=(,KEEP,DELETE),SPACE=(TRK,0),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         DCB=($DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$BASE.TC) LIMIT (3) SCR)
//*
//STEP4    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=$HLQ..SBVPTF2(BVPTC$LNG),DISP=SHR
//SYSUT2   DD DSN=$INDUN..$BASE.TC(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,(15,5),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=1067,BLKSIZE=10674)
//
```

Restoration of the test database

PACTTAB module: '$prfj.TAB' job

This job executes the RSTA procedure, using as input the backup which was loaded on the disk in the STEP4 of the '$prfj.TC' job;

Execution JCL

```
//$PRFJ.TTAB JOB ($CCPT),'TABLES TEST FILES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//  JCLLIB ORDER=($BIBP)
//*********************************************************************
//*    VA PACTABLES 3.5                                               *
```

```
//*************************************************************
//*        LOADING PACTABLES FUNCTION'S TEST FILES            *
//*************************************************************
//*
//STEP1   EXEC BVPRSTA
//
```

ACBs compilation

PACTACB module: '$prjf.ACB' job

This step is constituted of a '$PRFJ.ACB' job which builds all the ACBs via which the Pactables function can be used in online mode.

Execution JCL

```
//$PRFJ.TACB JOB ($CCPT),'ACBGEN',CLASS=$CLASSJ,
//  MSGCLASS=$MSGCL
//ET010   EXEC ACBGEN,SOUT=$OUT,PSB='$PSBLIB',
//             DBD='$DBDLIB',ACB='$ACBLIB'
//*************************************************************
//*                COMPILATION OF ACB'S     - PACTABLES 3.5   *
//*************************************************************
//*
//*-----------------------------------------------------------*
//*    THIS JOB MUST BE EXECUTED AFTER LOADING DBDLIB AND     *
//*    PSBLIB (EITHER BY LOADING OF OBJECT MODULES OR BY      *
//*    COMPILATION OF DBD AND PSB).                           *
//*-----------------------------------------------------------*
//*
//G.SYSIN DD *
  BUILD DBD=(PACDTD$SUG,PACDTE$SUG,PACDTG$SUG)
  BUILD DBD=(PACDTU$SUG,PACDTV$SUG,PACDTZ$SUG)
  BUILD DBD=(PACDT1$SUG,PACDTB$SUG)
  BUILD PSB=(BVPP500,BVPP501,BVPP510,BVPP520)
  BUILD PSB=(BVPP530,BVPP540,BVPP550,BVPP560)
  BUILD PSB=(BVPP570,BVPP580,BVPP590,BVPP599)
  BUILD PSB=(BVPP600,BVPP601,BVPPLNK)
  BUILD PSB=(BVPP610,BVPP620)
  BUILD PSB=(PTA250$SUG,PTA300$SUG)
//*
//
```

Update of user parameters

The system is operational only if the user parameters have been entered in the Pactables Database.

Before any test, the user parameters must be updated in the TG file using the PMTA procedure. An initial general user code is provided for the installation phase and is found in the TG file:

```
'********SUPER    '
```

Additional operational information

Since tables are rather stable files undergoing few updates, the Pactables function does not provide a journal file.

However, update transactions can be retrieved from the standard IMS journal.

List of installed programs

TO2LPT module: '$prfj.LPTA' job

This job executes the LPTA procedure that prints the list of batch and on-line programs with their compilation date.

This list must be kept since you may be asked for installation references by IBM in case of a problem in the operation of Pactables.

```
//$PRFJ.LPTA JOB ($CCPT),'PROGR.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*            --- LIST OF INSTALLED PROGRAMS  ---                    *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//LPTA     EXEC BVPLPTA
```

Use tests

The tests contain the following steps:

- On-line use tests,
- Batch tests for update, printing and reorganization,
- Tests for tables generation.

The test case contains 3 tables:

- 'TEMPER' without historical account,
- 'CUSTOM' with historical accounts dated 03/01/85 and 03/10/85.
- 'ARTICL' with historical account dated 01/15/87

On-line tests of the Pactables function:

- Open Pactables test files,
- Read-only access to all screens,
- Perform some updates.

Batch tests:

- Execute the PRTA procedure.

- Execute the EXTA procedure.
- Close Pactables files.
- Execute the UPTA procedure.

Reorganization of test tables:
- Save (IDCAMS) TV and TD.
- Execute the reorganization (RETA) which contains:
  - Reorganization of TV (PTA400 and PTA410 programs)
  - Reorganization of TD (PTA420 program)
  - Building of TC backup file (PTA430 program)
- Restore TV and TD (RSTA)
  - Execute a printing of the tables (PRTA) for verification.
  - Open the files and perform some on-line reorganization validation tests.

Test for table generation (GETT procedure)
- Close the files.
- Execute extraction with VA Pac (GETA or GETD).
- Execute GETT.
- Verify the execution.
- Re-open the files and perform some verification tests.

Test JCL: INTA

```
//$PRFJ.INTA JOB ($CCPT),'INIT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*           --- TEST OF THE INTA PROCEDURE ---                      *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//INTA     EXEC BVPINTA
I.B.M. ESSAI                        F 1234567  ABC
```

Test JCL: GETT

```
//$PRFJ.GETT JOB ($CCPT),'GENERATION',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*            --- TEST OF THE GETT PROCEDURE ---                     *
```

```
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//**** INSERT HERE GETA OR GETD (SEE VA PACBASE)
//GETT     EXEC BVPGETT,MD='&&MD'
```

Test JCL: PRTA

```
//$PRFJ.PRTA JOB ($CCPT),'PRINT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*            --- TEST OF THE PRTA PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//PRTA     EXEC BVPPRTA
 *********SUPER
EACUSTOM03101985
EATEMPER
```

Test JCL: IMTA

```
//$PRFJ.IMTA JOB ($CCPT),'IMPORT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*          --- TEST OF THE IMTA PROCEDURE ---                       *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//IMTA     EXEC BVPIMTA,TABF='...'
 *********SUPER
 A??????
```

Test JCL: UPTA

```
//$PRFJ.UPTA JOB ($CCPT),'UPDATE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*           --- TEST OF THE UPTA PROCEDURE ---                      *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//UPTA     EXEC BVPUPTA
 *********SUPER
 ACUSTOM03101985  *
AV 44190
 V 5555333***ATHENS*
 V 6666333***MOSCOW*
 V 8899000***PEKING*
AV 3333111
 V 6666111*MARAVEN*BOLIVAR*CARACAS*22300*VENEZUELA*3
```

```
 ATEMPER        /
 V GUAYAQUIL/0F 75 0C 24/0F 75 0C 24/0F 78 0C 25/0F 78 0C 25
 V-/0F 78 0C 25
 V-/0F 78 0C 25/0F 80 0C 27/0F 80 0C 27/0F 78 0C 25/0F 78 0C 25
 V-/0F 78 0C 25/0F 75 0C 24/
 V PARIS/////0F 58 0C 14/
 V FRANKFURT/0F 30 0C -1/0F 32 0C  0/0F 39 0C  4/0F 46 0C  7
 V-/0F 55 0C 13
 V-/0F 60 0C 15/0F 64 0C 18/0F 63 0C 17/0F 57 0C 14/0F 48 0C  9
 V-/0F 38 0C  4
 V-/0F 33 0C 1/
AV FRANKFORT
```

Test JCL: SVTA

```
//$PRFJ.SVTA JOB ($CCPT),'SVTA',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*            --- TEST OF THE SVTA PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//SVTA     EXEC BVPSVTA
```

Test JCL: RSTA

```
//$PRFJ.RSTA JOB ($CCPT),'RSTA',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*           --- TEST OF THE RSTA PROCEDURE ---                      *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//RSTA     EXEC BVPRSTA
```

Test JCL: RETA

```
//$PRFJ.RETA JOB ($CCPT),'REORG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*            --- TEST OF THE RETA PROCEDURE ---                     *

//*********************************************************************
// JCLLIB ORDER=($BIBP)
//RETA     EXEC BVPRETA
 *********SUPER
GACUSTOM03101985
GATEMPER
```

Test JCL: PMTA

```
//$PRFJ.PMTA JOB ($CCPT),'PARAM.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*          --- TEST OF THE PMTA PROCEDURE ---                       *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//PMTA     EXEC BVPPMTA
 ********TASUPER
 ********TJ000100//$PRFJ.PRTA JOB ($CCPT),'PRTA',CLASS=$CLASSJ,
 ********TJ000200//           MSGCLASS=$MSGCL
 ********TJ000300//PRTA  EXEC BVPPRTA
 USER1   TAUSER1   2
```

Test JCL: EXTA

```
//$PRFJ.EXTA JOB ($CCPT),'EXTRACTION',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*            --- TEST OF THE EXTA PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//EXTA     EXEC BVPEXTA
 *********SUPER
 ACUSTOM03101985
/*
//* EXTRACTED TRANSACTIONS FILE
//PTA160.PAC7NU DD DSN=---.---.---,DISP=SHR
```

Test JCL: TUTA

```
//$PRFJ.TUTA JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*          --- TEST OF THE TUTA PROCEDURE ---                       *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//TUTA     EXEC BVPTUTA
 *********SUPER
 ACUSTOM10311985
 ATEMPER
```

Test JCL: CDT1 (DTM)

```
//$PRFJ.CDT1 JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*             --- TEST OF THE CDT1 PROCEDURE ---                    *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//CDT1     EXEC BVPCDT1,
//* TDMAST = 'MASTER' TABLE-DESCRIPTION FILE
//* TDSLAV = 'SLAVE' TABLE-DESCRIPTION FILE
//* XD     = EXTRACTED-DESCRIPTION FILE
// TDMAST=---.---.---,TDSLAV=---.---.---,XD=---.---.---
 *********SUPER
 ACUSTOM
 ATEMPER
/*
```

Test JCL: CDT2 (DTM)

```
//$PRFJ.CDT2 JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*             --- TEST OF THE CDT2 PROCEDURE ---                    *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//CDT2     EXEC BVPCDT2,
//* TDSLAV = 'SLAVE' TABLE-DESCRIPTION FILE
//* TUSLAV = TV INDEX
//* TVSLAV = TABLES ASSOCIATED TO 'SLAVE' DESCRIPTION
//* XD     = FILE OF TABLE-DESCRIPTIONS EXTRACTED IN CDT1
//  TDSLAV=---.---.---,TVSLAV=---.---.---,TUSLAV=---.---.---,XD=-
```

Test JCL: CVTA (DTM)

```
//$PRFJ.CVTA JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VA PACTABLES 3.5                                                  *
//*********************************************************************
//*           --- TEST OF THE CVTA PROCEDURE ---                      *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//CVTA     EXEC BVPCVTA,
//* TD = TABLE-DESCRIPTION FILE
//* TU = INDEX OF TV FILE
//* TV = FILE OF TABLES ASSOCIATED TO DESCRIPTIONS
//  TD=---.---.---,
```

```
//  TU=---.---.---,
//  TV=---.---.---
 *********SUPER
SACUSTOM31101985    31101987
```

Pactables standard reinstallation

Pactables must be re-installed when a new version of the software comes out following corrections and enhancements.

To install this version, download the cartridge in the dedicated PDS by using SMP/E and execute the JCL supplied if it is necessary.

A sub-release, identified by a number is composed of:

- An installation cartridge (or tape),
- The "Program Directory for Pactables" specific to SMP/E,
- The list of corrected anomalies,
- Instructions -- possibly included -- to complete this chapter.

Generally, only system files and program libraries are impacted by this version.

In any case, load-modules are updated by SMP/E. They are copied into the hlq.SBVPMBR8 and hlq.SBVPMTR8 PDSs.

If a JCL is modified, run the BVPMMJCL utility again with the parameters provided at installation, and split the result file to re-install the JCLs as described in the Installation chapter.

If the error messages file is modified, run the PACTTE JCL again.

Chapter 6. Pactables - RACF or TOPSECRET Interface

Introduction

Security systems provide a mechanism for data access control. They perform user identification and verification, and they check resource access authorizations.

These systems control security over the whole data processing installation and, as such, operate independently from the Pactables system. The Interface is designed to ensure control communication between the security systems and Pactables. The Security Interface performs the following tasks:

- On-line: automatic retrieval of the CICS or IMS SIGN-ON USERID, that is displayed on Pactables sign-on screen.
- In batch: for the Pactables procedures which include user input ('*' line) and are executed under TSO, the user code and password can then be left blank.

In order to ensure compatibility with all security systems, Pactables is not directly connected with the security system, but with SAF (System Authorization Facility) via the RACROUTE macro-instructions for RACF or the TSS macro-instructions for TOPSECRET.

Installation

1. Class creation

Each Pactables Database must be identifiable to the security system. As a result, each database corresponds to a class. A class must be created within the Pactables function by the Pactables manager using the '$TRANP' transaction, and:

- under RACF via the 'ICHERCDE' RACF macro.
- under TOPSECRET via the command:

```
TSS RESCLASS(cccc) RESCODE(xx), with
  cccc = resource class,
  xx = hexadecimal code which identifies the resource.
```

A class name is coded on four characters and must be identical in the security system and in Pactables.

2. Resource creation

The creation of resources is necessary only when these are going to be checked by the security system.

A class includes all the logical resources of a Pactables Database, i.e. all the possible access authorizations for each table (down to the subschema, subsystem levels). These authorizations include the following data:

AUTHORIZATION + SUB-SCHEMA + SUB-SYSTEM + TABLE NUMBER

The authorization search is processed according to the order of these elements. If there are no sub-schema, sub-system and table number, blanks are replaced by '$' signs. When a table is not assigned a specific authorization, the general access authorization is taken into account.

For RACF:

Resources are created via the 'RDEFINE' procedure.

For TOPSECRET, resources are created via the command:

```
TSS ADD (dept-name) cccc(nstable) cccc(nstable)... with
  dept-name = department name,
  n = sub-schema number,
  s = sub-system number,
  table = table code.
```

EXAMPLE:

The Pactables manager wishes to check all table authorizations on a given table:

```
SUB-SCHEMA NUMBER      SUB-SYSTEM NUMBER        TABLE NUMBER
      1                        3                    TABLE
```

The search is performed in the following order:

```
 1   1    3      TABLE
 2   $    3      TABLE
 3   1    $      TABLE
 4   $    $      TABLE
 5   $    $      $$$$$$
```

Under RACF or TOPSECRET, the asterisk is a generic character. As a result, on the sites controlled by a security system, the Pactables manager's code is '$$$$$$$$'.

3. Defining users

Resource access authorizations are granted to individual end users :

- under RACF via the 'PERMIT' procedure.
- under TOPSECRET via the command:

```
TSS PERMIT (user-code) cccc(nstable), with
  cccc = resource class,
  n = sub-schema number,
  s = sub-system  number,
  table = table code.
```

Resources and user codes not declared to the security system are consequently prohibited in Pactables operations.

4. Batch and on-line requests

You must execute the RACF request via an assembler sub-program: BVPSECUR.

This sub-program is provided as BVPSECRA in the hlq.SBVPMBR8 batch load-modules library.

Moreover, you must install an SVC in the LPA library.

To do so, you must:

1. Declare the SVC with the appropriate routine number.

Example for number 232: in SYS1.PARMLIB(IEASVC00), add the line SVCPARM 232,REPLACE,TYPE(3),EPNAME(IGC0023B) where IGC0023B is the name of the SVC load module.

2. Rename BVPSECRA into BVPSECUR and then link the BVPSECUR object module to create the appropriate load module name.

3. Add this load module name into the SVC load system library.

4. ZAP the BVPSECB and BVPSECT VA Pac modules with the chosen SVC routine number.

5. After the LPA library update, you must execute an IPL CLPA to take the modifications into account.

The E10RACF sample JCL, that is supplied with the installation JCLs, performs steps 2 and 3, and the E11RACF JCL executes step 4.

If the SVC has already been installed for VA Pac, you just have to execute the E11RACF JCL.

Then, you must check whether the user is authorized to run the procedure.

Operating mode

The acquisition of the Pactables / RACF or TOPSECRET interface requires a modification of the database parameters. The '$TRANP' transaction enables the database manager to update these parameters, specifying the security system type ('R' for RACF or 'S' for TOPSECRET), the Pactables database identification class, and two indicators:

- User authorization indicator:

  For RACF, it specifies whether the user, connected to CICS or IMS for the on-line mode, or TSO for the batch mode, is allowed a Pactables connection under a user code other than his/hers. This indicator applies only when the security system is used.

  For TOPSECRET, this indicator is forced because the user cannot log on with a user code other than his/hers.

- Resource indicator:

  Access through Pactables or through the security system. This indicator applies only when the security system is used.

These indicators are used to distinguish the following methods of management: total and strict management or total and flexible management under a security system.

Total and strict management under a security system

Verification of users and table access is managed by the security interface, and a user can log on only with his/her own code.

1. Logging on in on-line mode: the PACTABLES SIGN-ON screen is initialized with the code under which the user logged on to CICS or IMS. This code is retrieved in the IO-PCB under IMS and by an EXEC CICS ASSIGN USERID command under CICS (valid only from CICS release 1.7 on). Changing the user code is prohibited.
2. The password field is locked and cannot be filled in. The cursor is positioned on the library code.
3. RACF only: LJ and LE screens: since RACF does not carry over the user code and the CICS or IMS password, they must be inserted on the JOB card. Since the system does not pass on the password, the user must enter this on the LJ or LE screen (masked fields) when first submitting a JOB or SUB action.

   A warning message is displayed if the field has not already been filled in. From RACF V1.9, the password no longer needs to be filled in since a user can submit a job for another ('surrogate') user.

4. Batch procedures that include a '*' line: the user code and password are no longer required since the system automatically takes the code under which the user signed on to TSO. As a result, the PASSWORD is no longer present in the temporary files found in batch jobstreams.

   For RACF only: another consequence is that jobstreams including steps with a '*' line can be linked together without manual intervention so that the password can be specified. This process implies a restriction: the user cannot code several '*' lines with user codes other than his/her own for procedures which would normally allow him/her to do so.

   Note: with TOPSECRET, the user can never enter a code other than his/her own.

Total and flexible management under a security system

This management is possible with RACF only.

Verification of users and library access is managed by the security interface, but the user can log on with a code other than his/her own.

1. Logging on in on-line mode: identical to number 1 under 'TOTAL AND STRICT MANAGEMENT UNDER A SECURITY SYSTEM' above, but the field including the USER code is an input field, as is the PASSWORD field. The user can modify these two fields (password is required). In case of modification, the interface performs a test to validate the USER code, and the security system performs a test to validate the password.
2. LJ, LE screens: identical to number 3 under 'TOTAL AND STRICT MANAGEMENT UNDER A SECURITY SYSTEM' above. If the user entered the password on the logon screen, it does not have to be entered again.
3. Batch procedures that include a '*' line: just as in the case of on-line processing when the user code is different from the one under TSO, the password has to be filled in. This makes it possible to submit jobs with several '*' lines having different user codes.

   Temporary files do not include the password, which means that it is not possible to link together steps having a '*' line. The password has to be entered each time.

   Nevertheless, if the USER code is identical to the one under TSO, verification of users and library access is managed as described in number 4 under 'TOTAL AND STRICT MANAGEMENT UNDER A SECURITY SYSTEM' above.

The 'TYPE' field of the $TRANP transaction can therefore contain one of the following two values: "blank" or "P". "P" stands for the resource verification by Pactables and not by the security system.

The 'BLOC' field has one of the following two values: "blank" or "N". "N" specifies that the user cannot use a password other than his/her own.

Part Number: DETIM000351A - 7426

Printed in USA