VisualAge Pacbase 2.5

PACTABLES 2.5 – IBM MVS, IMS
OPERATIONS MANUAL

DETIM000252A

Note

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

According to your license agreement, 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.software.ibm.com/ad/vapacbase/support.htm

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

Second Edition (August 1999)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:
http://www.software.ibm.com/ad/vapacbase/support.htm
or to the following postal address:
IBM Paris Laboratory
VisualAge Pacbase Support
30, rue du Château des Rentiers
75640 PARIS Cedex 13
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.

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:

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

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 information which has been exchanged, should contact:

IBM Paris Laboratory
SMC Department
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

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.

TABLE OF CONTENTS

1. FOREWORD

FOREWORD

The purpose of this manual is to provide the reader with information related to the installation and operation of Pactables 2.5.

HOW TO USE THIS MANUAL FOR SYSTEM INSTALLATION

If a previous Pactables Release is already installed on the site:

. The 2.5 Release must be different from any former Pactables Release regarding installation parameters. The test case provided on the installation tape must be executed.

. Once the installation is complete, read the chapter about the retrieval and follow the instructions carefully in order to ensure a thorough compatibility with the former release existing data.

2. PACTABLES COMPONENTS

2.1. INTRODUCTION

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 function, 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:

The installation of the Pactables function is quite independent of that of 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 2.5 Operations Manual.

Options of the Pactables function are coded as follows:

. Dispatched Table Management : DTM

. Security System Interface : SEC (only with IBM MVS)

2.2. THE ON-LINE PROGRAMS LIBRARY

THE ON-LINE PROGRAMS LIBRARY: MTR8

Its size is approximately 20 tracks (3380 disk unit).

The table below lists these programs.

CODE	OPERATION AND MEANING
$ROOT.FT00	Table extraction (TUF-TP facility)
$ROOT.FT90	User Interface (TUF-TP facility)
$ROOT.P500	Initial screen
$ROOT.P501	Initial screen preparation
$ROOT.P510	C1: read-write access to mono-item
	with CR, MO, DE for updating
$ROOT.P520	C2: read-only access to multi-item
	with DE for an item deletion
$ROOT.P530	LT : list of tables
$ROOT.P540	LS : list of sub-schemas/sub-systems
$ROOT.P550	LD : list of documentation
$ROOT.P560	C3 : consultation of archived item
$ROOT.P570	C? : 'HELP' screen
$ROOT.P580	LH : list of historical accounts
$ROOT.P590	LJ LE : JOB function print requests
$ROOT.P599	Display system errors
$ROOT.P600	Update passwords and language
$ROOT.P601	Password/Language code screen preparation
$ROOT.P610	User code updating ('DTM' option)
$ROOT.P620	Access authorization updating ('DTM' option)
$ROOT.PLNK	Access to the Pactables transaction by user transactions
$ROOT.R980	Message formatting sub-program
PACXSY	OS/VS1 - MVS 'JOB' function
MVSJOBB	MVS 'JOB' function
PACSECB	Security System sub-program (SEC extension)

$ROOT = 2-character prefix for Pactables.
Example: P2P510 ($ROOT='P2').

IMPORTANT

Two additional programs (P512 and P522) are supplied since Pactables 2.0.

During updates, the P510 and P520 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.

With Pactables 2.0 and higher, if the user check routines are generated with the century management option, the two new programs (P512 and P522) must be renamed and used instead of P510 and P520.

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

2.3. THE BATCH PROGRAMS LIBRARY

THE BATCH PROGRAMS LIBRARY: MBR8

Its size is approximately 35 tracks (3380 disk-unit).

The table below lists these programs.

CODE	PROC.	OPT.	MEANING
PTU001	all		Include input transactions
PTAD05	CDT1	DTM	DTM option: Table descr. comparison
PTAD10	--	DTM	-- -- -- --
PTAD20	CDT2	DTM	DTM option: Table descr. updating
PTAINI	INTA		TD and TV db's initialization
PTARSD	RSTA		TD database restoration
PTARSV	-		TV database restoration
PTARSG	-		TG database restoration
PTAR20	RTTA		Retrieval Pactables 8.02/1.2 -> 2.0
	R3TA		Retrieval Pactables 7.3
PTABVS	BVTA		Copy TD, TV base onto VSAM files
PTASTD	GETA		Table generation
PTASVD	SVTA		TD database backup
PTASVV	-		TV database backup
PTASVG	-		TG database backup
PTAT10	LDTE		Load TE Error Message database
PTAT15	LDTG		Load TG User Codes database
PTAT17	LDTZ		Load TZ SPA database
PTAU80	TUTA		Direct consultation of tables
PTAV10	CVTA	DTM	DTM option: Table comparison
PTAV20	--	DTM	-- -- --
PTAXDT	LPTA		List of installed modules
PTAXVD	R3TA		Retrieval Pactables 7.3
PTAXVV	R3TA		-- -- --
PTAXVG	R3TA		-- -- --
PTA100	PMTA		Update of user codes
PTA120	--		-- --
PTA150	EXTA		Extract TABLES (MJTA trans.)
PTA160	--		-- -- --
PTA250	GETT		Table generation
PTA290	--		Table generation / list printing
-	LDTA		-- -- --
PTA300	UPTA		Table updating
PTA310	IMTA		Table import
PTA320	PRTA		Table printing
PTA350	UPTA		Table updating / printing
-	PRTA		-- -- --
PTA360	UPTA		Table updating / printing
-	PRTA		-- -- --

BATCH Programs (cont'd)

CODE	PROC.	OPT.	MEANING
PTA400	RETA		Table reorganization
PTA410	--		-- -- --
PTA420	--		-- -- --
PTA430	--		-- -- --
PAP822	sub-pgm		Optimized table access
PAP820	-		Optimized table access (for COBOL VS environment) (2)
PAP922	-		Table access by user program
PAP920	-		Table access by user program (for COBOL VS environment) (2)
PACSECB	-		Security System (3)
PTATCD	TCTA		Sort of TD file
PTATCG	-		Sort of TG file
PTATCV	-		Sort of TV file
PTATC1	-		Sort TC according to type of file
PTATC2	-		Reconstitution of sorted TC

NOTES

(2) These access programs are to be used in a COBOL VS user environment.

SECURITY SYSTEMS INTERFACE EXTENSION (SEC)

The optional sub-program PACSECB interfaces the Pactables function with the site's security system when this extension has been selected.

This sub-program is found on the tape in the batch load-module library (PACT.MBR8). It must be installed in an authoized library (see in Chapter INSTALLATION, subchapter 'Installation Process', the section about the '$prfj.TSYS' job.)

Refer to the VISUALAGE PACBASE SECURITY SYSTEMS INTERFACE manual for more information on how to use this extension.

IMPORTANT NOTE:

Two other programs (PTA302 and PTA312) are supplied since release 2.0.

During updates, the PTA302 and PTA312 programs may call the user check routines in order to perform additional checks. The default generation option of these routines is 'without century management'.

With Pactables 2.0 and higher, if the user check routines are generated with the century-management option, the two new programs, PTA302 and PTA312, must be renamed and used respectively in the UPTA and IMTA procedures instead of the PTA300 and PTA310 programs.
In all cases, ALL the user check routines should be generated with the same century-management option.

2.4. THE VA PAC MACRO-STRUCTURES LIBRARY

THE TUF-TP MACRO-STRUCTURES LIBRARY

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
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 the call of xxFT90 sub-program in the TUF-TP facility.

These Macro-structures are supplied as VA Pac update transactions. They must be loaded in the VA Pac library used for the development of user transactions by taking the transactions of VA Pac UPDT procedure in input.

2.5. THE PARAMETERS LIBRARY

THE PARAMETER LIBRARY: SY

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

.The DEFINE and VERIFY of VSAM files:

The DELETE/DEFINE of each of Pactables VSAM file is named using the following convention:

DF$ROOT.$FILE.xx (with xx=AD, AV, DD, TD, TG, TU, TV, TS, TW, T1 and TB), and DF$ROOT.$ROOT.xx (with xx = TE, TZ),

Example: DFP200TD or DFP2P2TE.

The VERIFYs of each of the VSAM files of the Pactables function are named as VERIFxx

($ROOT = 2-character prefix, $FILE = 2-character database number.)

.The DL/1 Source files:

```
PACDxx$SUF  : DBDs with xx=TD, TE, TG, TU, TV, TZ, T1, TB.
              $SUF=2 characters (DBD suffix):
              Example: PACDTV22 ($SUF = '22').
PTAyyy$SUG  : PSBs of the batch programs, with yyy=last
              3 characters of the program codes,
              $SUG = 2 characters (Batch PSB suffix):
              Example: PTA15022  ($SUG = '22').
$ROOT.Pzzz  : PSBs of the on-line programs, with zzz=last
              3 characters of the program codes,
              $ROOT = 2 character-prefix:
              Example: P2P510  ($ROOT = 'P2').
(See the list of batch and on-line programs.)
```

.The initial record for the TG, TZ and TB databases:

```
LD$ROOT.$FILE.TG
LD$ROOT.$ROOT.TZ
LD$ROOT.$FILE.TB
```

.The SYSIN used for TG loading:

```
SY$ROOT.$FILE.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$ROOT.$FILE.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 Macro-instructions 'APPLCTN' and 'TRANSACT':

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.

2.6. THE SYSTEM FILES

THE 'SYSTEM' FILES AND DATABASES

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:  (MBR8)

      .Size    : 35 tracks (3380 disk unit)
      .Blksize : 6,144

.The library of on-line load modules: (MTR8)

      .Size    : 20 tracks (3380 disk unit)
      .Blksize : 6,144

.The library of parameters (PDS) SY:

      .Size    : 05 tracks (3380 disk unit)
      .Blksize : 6,080

.The database containing the error messages and auto-
 matic HELP documentation of the Pactables function
 (TE):

      .Size         : Approx. 1,000 records
      .Organization : DL/1 HISAM
      .DSN          : $INDEX..$ROOT.$ROOT.TE
      .DBD          : PACDTE$SUF
      .Record len.  : 98 bytes
      .Segment len. : 90 bytes

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

      .Size         : Approx. 700 records
      .Organization : Sequential
      .DSN          : $INDEXP..$ROOT.$ROOT.IN
      .LRECL        : 90 bytes
      .BLKSIZE      : 1,800 bytes


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

       a DBD library      (DBDLIB)
       a PSB library      (PSBLIB)
```

THE DBD LIBRARY

```
      .Organization : PDS
      .DSN          : (chosen at installation)
      .DCB          : (RECFM=U,BLKSIZE=6,144)
      .Size         : 01 track (3380 disk unit)
```

THE PSB LIBRARY

```
      .Organization : PDS
      .DSN          : (chosen at installation)
      .DCB          : (RECFM=U,BLKSIZE=6,144)
      .Size         : 06 tracks (3380 disk-unit)
```

2.7. THE EVOLVING FILES

EVOLVING FILES AND DATABASES

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

Pactables Database (TD Database: Table Descriptions)

.Organization : DL/1 HISAM Database
.DSN : $INDEX..$ROOT.$FILE.TD
.DBD : PACDTD$SUF
.Segment lgth : 240 bytes
.Record lgth : 248 bytes
.Space : 16 records per C.I. of 4,096

Pactables Database (TU: Primary Index of TV)

.Organization : TV VSAM INDEX
.DSN : $INDEX..$ROOT.$FILE.TU
.DBD : PACDTU$SUF
.Segment lgth : 35 bytes
.Record lgth : 40 bytes
.Space : 25 records per C.I. of 1,024

Pactables Database (TV Database: Table contents)

.Organization : DL/1 HIDAM Database
.DSN : $INDEX..$ROOT.$FILE.TV
.DBD : PACDTV$SUF
.Segment lgth : 80 to 1,100 bytes
.Record lgth : 4,089 bytes
.Space : according to the length of the tables

Pactables Database (TG Database: User codes)

.Organization : DL/1 HISAM Database
.DSN : $INDEX..$ROOT.$FILE.TG
.DBD : PACDTG$SUF
.Segment lgth : 85 bytes
.Record lgth : 94 bytes
.Space : 43 records per C.I. of 4,096.

Pactables Database (SPAs' TZ Database)

.Organization : DL/1 HISAM Database .DSN : $INDEX..$ROOT.$ROOT.TZ .DBD : PACDTZ$SUF .Segment lgth : 8,500 bytes .Record lgth : 8,510 bytes .Space : 1 record per C.I. of 10,240

Pactables (T1 Database TB primary index)

.Organization : VSAM TB index .DSN : $INDEX..$ROOT.$FILE.T1 .DBD : PACDT1$SUF .Segment lgth : 63 bytes .Record lgth : 68 bytes .Space : 15 records per C.I. of 1,024

Pactables (TB Database working file TUF-TP)

.Organization : DL/1 HIDAM Database .DSN : $INDEX..$ROOT.$FILE.TB .DBD : PACDTB$SUF .Segment lgth : 80 to 1,140 bytes .Record lgth : 4,089 bytes .Space : according to the length of the tables.

Pactables Database Backup File (TD, TV, TG)

.Organization : Sequential with generation .DSN : $INDEXQ..$ROOT.$FILE.TC .DCB : (RECFM=VB,LRECL=1,067,BLKsize=10,674) .Space : 240 bytes for the decriptions (TD) : 80/1,061 bytes for the contents (TV)

This is a standardized Pactables backup file in 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

3. ENVIRONMENT

3.1. INTRODUCTION

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 Chapter ENVIRONMENT of the VisualAge Pacbase INSTALLATION AND ENVIRONMENT Operations Manual.

3.2. ON-LINE ENVIRONMENT

ON-LINE ENVIRONMENT

The Monitor used for TP Pactables is IMS/DC.

Since Version 2.5 of Pactables, the MFS system is bypassed by the use of the DFS.EDTN standard module for the receiving and display of messages.

The formatting of messages is managed by the xxR980 sub-program found in the on-line programs library.

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

- $TRANT transaction code of connection to Pactables,

- $TRANP transaction code of Users management.

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

3.3. ACCESS METHODS

GENERAL PRINCIPLES

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

Thus:

. The user cannot directly access Table data via DL/1 standard utilities. This can be done only via utilities provided with Pactables.

. DL/1 resources are seldom used by the Pactables func- tion. 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 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.

3.4. BATCH ENVIRONMENT

BATCH ENVIRONMENT

In batch mode, the running of 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 buffers allocated to the files used by these procedures.

4. THE BATCH PROCEDURES

4.1. INTRODUCTION

INTRODUCTION TO THE BATCH PROCEDURES

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:

- symbolics or parameters used,
- list of the files used (temporary or permanent),
- return codes for each step

. JCL lines.

4.2. CLASSIFICATION OF PROCEDURES

CLASSIFICATION OF PROCEDURES

The batch procedures are the following:

.Pactables file initialization (INTA)

.Transfer of TD-TV databases in 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)

With the Dispatched Table Management option (DTM):

.Table description comparison (CDT1, CDT2)

.Table extraction for update (CVTA).

For retrieval of previous releases:

.Retrieval of 7.2 Pactables files (R2TA)

.Retrieval of 7.3 Pactables files (R3TA)

.Retrieval of the 8.xx or 1.2 backup file (RTTA).
(not for IMS).

NOTE

Pactables does not provide a journal of update transactions.

4.3. ABNORMAL EXECUTIONS

ABNORMAL EXECUTIONS

Generally, all batch programs may end abnormally. Specific- ally, 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.

5. TABLE INITIALIZATION (INTA)

5.1. INTRODUCTION

INTA: TABLE INITIALIZATION

INTRODUCTION

This procedure initializes the tables descriptions and contents files of the Pactables Database.

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 'TABLE GENERATION' for more details on the Table initialization procedure).

EXECUTION CONDITION

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

5.2. USER INPUT

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

5.3. DESCRIPTION OF STEPS

INTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


DEFINITION OF FILES:  IDCAMS

 -Tables descriptions file :            PAC7TD
  DSN=&INDEX..&ROOT.&FILE.TD
 -Tables contents file    :           PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU


INITIALIZATION OF FILES:  PTAINI

.Input file
  PAC7MD

.Output files:
 -Tables Descriptions File
  DSN=&INDEX..&ROOT.FILE.TD
 -Table Contents File : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.FILE.TV
  DSN=&INDEX..&ROOT.FILE.TU

.Output report:
 -Initialization review
  PAC7ED
```

5.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*             - INITIALIZATION OF TABLE MANAGEMENT FILE -           *
//*********************************************************************
//$RADP.INTA   PROC ROOT=$ROOT,        ROOT OF VA PAC SYSTEM
//             FILE=$FILE,             NUMBER OF PHYSICAL DATABASE
//             INDEX='$INDEX',         VSAM INDEX
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//             SPAMB='(TRK,(1,1),RLSE)', SPACE MB FILE
//             OUT='$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=PTU001
//*********************
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PAC7MB  DD DSN=&&INTAMB,DISP=(,PASS),UNIT=$UWK,
//           DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE   DD DDNAME=SYSIN,DCB=BLKSIZE=80
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TV),DISP=SHR
//PTAINI  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAINI,PTAINI$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7MD   DD DSN=&&INTAMB,DISP=SHR
//PAC7TD$SUG DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TU$SUG DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUG DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7ED   DD SYSOUT=&OUT
//*
```

6. TD-TV DATABASES ON VSAM FILES (BVTA)

6.1. INTRODUCTION

COPY OF TD-TV DATABASES ONTO VSAM FILES (BVTA)

This utility function executes the PTABVS program used to copy the TD and TV DL/1 databases onto VSAM files.

EXECUTION CONDITION

None since the database is not updated.

6.2. DESCRIPTION OF STEPS

GETT: DESCRIPTION OF STEPS

```
DEFINITION OF FILES:  IDCAMS

 -Tables descriptions file :            PAC7TS
  DSN=&INDEX..&ROOT.&FILE.TS
 -Tables contents file :                PAC7TW
  DSN=&INDEX..&ROOT.&FILE.TW


VERIFICATION OF VSAM FILES:  IDCAMS


COPY OF DATABASES ONTO VSAM FILES: PTABVS

.Input files:

 -Tables descriptions file :         PAC7TD
  DSN=&INDEX..&ROOT.&FILE.TD
 -Fichier des contenus de tables :  PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU

.Output files:

 -Tables descriptions file :         PAC7TS
  DSN=&INDEX..&ROOT.&FILE.TS
 -Tables contents file :             PAC7TW
  DSN=&INDEX..&ROOT.&FILE.TW
```

6.3. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*       - COPY OF TD-TV DATABASES ONTO TS-TW VSAM FILES-            *
//*********************************************************************
//$RADP.BVTA   PROC FILE=$FILE,    PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,         VA PAC SYSTEM ROOT
//             INDEX='$INDEX',     VSAM FILE INDEX
//             INDEXP='$INDEXP',   NON-VSAM FILE INDEX
//             STEPLIB='$MODB',    BATCH PROGRAM LIBRARY
//*:          VSAMCAT='$CATU',    USER VSAM CATALOG
//             OUT='$OUT',         OUTPUT CLASS
//             RESLIB='$RESLIB',   IMS RESLIB
//             PROCLIB='$PRCLIB',  IMS PROCLIB
//             DBDLIB='$DBDLIB',   DBD LIBRARY
//             PSBLIB='$PSBLIB',   PSB LIBRARY
//             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=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TW),DISP=SHR
//      DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TS),DISP=SHR
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTABVS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTABVS,PTABVS$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7TS   DD DSN=&INDEX..&ROOT.&FILE.TS,DISP=SHR
//PAC7TW   DD DSN=&INDEX..&ROOT.&FILE.TW,DISP=SHR
//*
```

7. TABLE GENERATION (GETT)

7.1. INTRODUCTION

GETT: TABLE GENERATION

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 in input by 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 DLIBATCH.

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.

7.2. DESCRIPTION OF STEPS

GETT: DESCRIPTION OF STEPS

```
VERIFICATION OF VSAM FILES:  IDCAMS


UPDATE OF TABLE FILES:  PTA250

.Permanent input-output files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Table-contents file :            PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
 -Update transactions
  PAC7MD : DSN=&MD

.Output file
  PAC7TK : DSN=&&DE

.Output report:
 -Input/output errors on files
  PAC7ET

.Sort files:
  SORTWK01, SORTWK02, SORTWK03


PRINTING OF DESCRIPTIONS:  PTA290

.Permanent input file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Transaction input file:
 -Print request
  PAC7TE : DSN=&&DE

.Output report:
 -Printout of descriptions
  PAC7ID
```

7.3. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*               - GENERATION OF TABLES (PACTABLES SITE) -           *
//*********************************************************************
//$RADP.GETT   PROC FILE=$FILE,        PHYSICAL DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM INDEX
//             INDEXP='$INDEXP',       NON-VSAM FILE INDEX
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//             SORTLIB='$BIBT',        SORT LIBRARY
//             OUT='$OUT',             OUTPUT CLASS
//             OUTL=$OUT,              OUTPUT CLASS
//             UWK=$UWK,               WORK UNIT
//             MD='$INDEX..$ROOT.$FILE.MD', GENERATED DESCRIPTION DSN
//             SPAWK='(TRK,(50,10),RLSE)',  SPACE OF WORK FILES
//             RESLIB='$RESLIB',       IMS RESLIB
//             PROCLIB='$PRCLIB',      IMS PROCLIB
//             DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//             PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//             BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//             CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTA250  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(BMP,PTA250,PTA250$SUG,&IN,&OUT1,
//        &OPT&SPIE&TEST&DIRCA,&PRLD,&STIMER,&CKPTID,
//        &PARDLI,&CPUTIME,&NBA,&OBA,&IMSID,&AGN)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.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=$REGSIZ,
//        PARM=(DLI,PTA290,PTA290$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE   DD DSN=&&DE,DISP=(OLD,PASS)
//PAC7ID   DD SYSOUT=&OUTL
//*
```

8. TABLE UPDATE (UPTA)

8.1. INTRODUCTION

UPTA: TABLE UPDATING

INTRODUCTION

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

EXECUTION CONDITION

The TV and TD Table files being updated by UPTA, 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, PTA302, is available with Pactables 2.0 and higher.

During updates, the PTA300 program may call the user check routines in order to perform additional checks. The default generation option for these routines is 'without century management'.

From Release 2.0 and higher, the user check routines are generated with the century-management option. The new program, PTA302, must therefore be renamed and used instead of the PTA300 program.

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

8.2. USER INPUT

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 update:

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

8.3. DESCRIPTION OF STEPS

UPTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


TABLE UPDATE: PTA300

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -User parameters file
  DSN=&INDEX..&ROOT.&FILE.TG

.Permanent input-output file:
 -Tables contents file       : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU

.Input transaction file:
 -Update transactions
  PAC7MS : DSN=&&UPTAMB

.Output file:
 -Print requests
  PAC7DE : DSN=&&TABLE      LRECL=80

.Output report:
 -Transaction review
  PAC7ET

.Work file:
 -Prepared transactions
  PAC7MT : DSN=&&MVT300
```

```
FORMATTING FOR PRINTING:  PTA350

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Tables contents files     : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU

.Input transaction file:
 -Print request
  PAC7DE : DSN=&&TABLE

.Output file:
 -Print file
  PAC7ET : DSN=&&SPOOL

.Output report:
 -Statistics on printing
  PAC7EX


PRINTING OF TABLES:  PTA360

.Permanent input file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Input Transaction file:
 -Print file
  PAC7ET : DSN=&&SPOOL

.Output report:
 -Printing of tables
  PAC7EY

.Sort files:
  SORTWK01, SORTWK02, SORTWK03.
```

8.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - TABLE UPDATE -                                   *
//*********************************************************************
//$RADP.UPTA   PROC FILE='$FILE',      PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             ROOT OF VA PAC SYSTEM
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       NON-VSAM FILE INDEX
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        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 FILE
//             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=CGIB,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=PTU001
//*
//STEPLIB  DD DSN=&STEPLIB,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
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTA300  EXEC PGM=DFSRRC00,REGION=1536K,
//        PARM=(BMP,PTA300,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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7MS   DD DSN=&&UPTAMB,DISP=(OLD,DELETE)
//PAC7DE   DD DSN=&&TABLE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAWK
//PAC7MT   DD DSN=&&MVT300,DISP=(NEW,DELETE),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=95,BLKSIZE=6175),
//            SPACE=&SPAWK
//PAC7ET   DD SYSOUT=&OUTL
//*
//PTA350  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA350,PTA350$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.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=$REGSIZ,
//        PARM=(DLI,PTA360,PTA360$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
//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=&INDEXP..&ROOT.&ROOT.SY(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$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7ET   DD DSN=&&SPOOL,DISP=(OLD,DELETE)
//PAC7EY   DD SYSOUT=&OUTL
//*
```

9. TABLE PRINTING (PRTA)

9.1. INTRODUCTION

PRTA: TABLE PRINTING

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 on-line: refer to the Pactables Reference Manual for more details on batch printing submission.

9.2. USER INPUT

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
		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

9.3. DESCRIPTION OF STEPS

PRTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


EXTRACTION OF TABLES FOR PRINTING:  PTA320

.Permanent input files:
 -Table-description File
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -Table-contents file            : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
 -User-parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Update transactions
  PAC7CA : DSN=&&PRTAMB

.Output file:
 -Print requests
  PAC7DE : DSN=&&TABLE      LRECL=80

.Output report:
 -Transaction review
  PAC7XE

PREPARATION FOR PRINTING:  PTA350

.Permanent input files:
 -Tables descriptions File
  DSN=&INDEX..&ROOT.&FILE.TD
 -Table-contents file            : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
```

```
.Input transaction file:
 -Print requests
  PAC7DE : DSN=&&TABLE

.Output file:
 -Print file
  PAC7ET : DSN=&&SPOOL

.Output report:
 -Statistics on printing
  PAC7EX


PRINTING OF TABLES:  PTA360

.Permanent input file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Input transaction file:
 -Print file
  PAC7ET : DSN=&&SPOOL

.Output report:
 -Printing of tables
  PAC7EY

.Sort files:
  SORTWK01, SORTWK02, SORTWK03
```

9.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - PRINTING OF TABLES -                             *
//*********************************************************************
//$RADP.PRTA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       NON-VSAM FILE INDEX
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        BATCH-PROGRAM LIBRARY
//             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 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=PTU001
//STEPLIB  DD DSN=&STEPLIB,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
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTA320  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA320,PTA320$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7CA   DD DSN=&&PRTAMB,DISP=(OLD,DELETE)
//PAC7DE   DD DSN=&&TABLE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAWK
//PAC7XE   DD SYSOUT=&OUTL
//*
//PTA350  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA350,PTA350$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.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=$REGSIZ,
//        PARM=(DLI,PTA360,PTA360$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
//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=&INDEXP..&ROOT.&ROOT.SY(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$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7ET   DD DSN=&&SPOOL,DISP=(OLD,DELETE)
```

```
//PAC7EY   DD SYSOUT=&OUTL
//*
```

10. TABLE IMPORT (IMTA)

10.1. INTRODUCTION

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 in 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 concurrency.

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, PTA312, is shipped with Pactables 2.0 and higher releases.

During updates, the PTA310 program may call user check routines in order to perform additional checks. The default generation option for these routines is 'without century management'.

With Release 2.0 and higher, if the user check routines are generated with the century-management option, the new program, PTA312, must therefore be renamed and used instead of the PTA310 program.

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

10.2. USER INPUT

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)

10.3. DESCRIPTION OF STEPS

IMTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


TABLE CHECK AND UPDATE:  PTA310

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -User-parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Permanent input-output file:
 -Table-contents file       : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU

.Input file:
 -Request transactions
  PAC7MV : DSN=&&IMTAMB
 -External table file
  PAC7NK : DSN=&TABF        LRECL=999

.Output file:
 -Print requests
  PAC7DE : DSN=&&TABLE      LRECL=80

.Output report:
 -Execution report
  PAC7ET

FORMATTING OF PRINTOUT:  PTA350

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Table-contents file      : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU

.Input transaction file:
 -Print requests
  PAC7DE : DSN=&&TABLE

.Output file:
 -Print file
  PAC7ET : DSN=&&SPOOL

.Output report:
 -Printing statistics
  PAC7EX


PRINTING:  PTA360
```

.Permanent input file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Input transaction file:
 -Print file
  PAC7ET : DSN=&&SPOOL

.Output report:
 -Table printout
  PAC7EY

.Sort files:

  SORTWK01, SORTWK02, SORTWK03.

10.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                  - TABLE IMPORT -                                 *
//*********************************************************************
//$RADP.IMTA   PROC FILE=$FILE,        PHYSICAL DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       NON-VSAM FILE INDEX
//             TABF=,                  DSN OF IMPORT FILE
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        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=PTU001
//STEPLIB  DD DSN=&STEPLIB,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
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTA310  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA310,PTA310$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7MV   DD DSN=&&IMTAMB,DISP=(OLD,PASS)
//PAC7NK   DD DSN=&TABF,DISP=OLD
//PAC7DE   DD DSN=&&TABLE,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAWK
//PAC7ET   DD SYSOUT=&OUTL
//*
//PTA350  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA350,PTA350$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.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=$REGSIZ,
//        PARM=(DLI,PTA360,PTA360$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
//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=&INDEXP..&ROOT.&ROOT.SY(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$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7ET   DD DSN=&&SPOOL,DISP=(OLD,DELETE)
//PAC7EY   DD SYSOUT=&OUTL
//*
```

11. TABLE REORGANIZATION (RETA)

11.1. INTRODUCTION

RETA: TABLE REORGANIZATION

INTRODUCTION

From the Pactables Database, this procedure rebuilds the backup file containing the new tables 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 numeric data signed positive (not avaible in previous releases).

EXECUTION CONDITION

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

11.2. USER INPUT

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	ttttt	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.

11.3. DESCRIPTION OF STEPS

RETA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


REORGANIZATION OF TABLE CONTENTS:  PTA400

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -Table-contents file      : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
 -User parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Reorganization requests
  PAC7DR : DSN=&&RETAMB

.Output file:
 -Reorganized-contents file
  PAC7TX : DSN=&&TXREO      LRECL=1063
 -Reorganized table list file
  PAC7DE : DSN=&&DEREO      LRECL=80
```

NOTE: This 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.

```
.Ouput report:
 -Transaction report
  PAC7IR

.Sort files:
  SORTWK01, SORTWK02, SORTWK03

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

VALIDIDATION OF TABLE CONTENTS:  PTA410

.Input transaction file
 -Reorganization requests
  PAC7MB : DSN=&&RETAMB

.Input file:
 -Reorganized contents file
  PAC7TX : DSN=&&TXREO

.Output file:
 -Validated contents file
  PAC7TW : DSN=&&TWREO
```

```
.Sort files:
  SORTWK01, SORTWK02, SORTWK03


REORGANIZATION OF TABLE-DESCRIPTIONS:  PTA420

.Permanent input file:
 -Tables descriptions file
 DSN=&INDEX..&ROOT.&FILE.TD

.Input file:
 -Reorganized table list file
  PAC7DE : DSN=&&DEREO

.Output files:
 -Reorganized tables descriptions file
  PAC7TS : DSN=&&TSREO
 -Tables descriptions print request
  PAC7ML : DSN=&&ME
```

NOTE: This file must be kept and used as input of the LDTA procedure, to produce a printout of the tables descriptions that were kept, in order to check the correct execution of the reorganization.

.Sort files -SORTWK01, SORTWK02, SORTWK03

```
BUILDING OF BACKUP FILE:  PTA430

.Input files:
 -Validated contents fille
  PAC7TW : DSN=&&TWREO
 -Reorganized descriptions file
  PAC7TS : DSN=&&TSREO

.Ouput file:
 -Backup file resulting from
  reorganization
  PAC7TC : DSN=&&INDUN..&ROOTT.00TC(+1)

TG FILE BACKUP:  PTASVG

.Permanent input file:
 -User-parameter file
  DSN=&INDEX..&ROOT.&FILE.TG
.Output file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=MOD
```

11.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - REORGANIZATION OF TABLES -                       *
//*********************************************************************
//$RADP.RETA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//             INDEXQ='$INDEXQ',       DATA GROUP FILE INDEX
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//             SORTLIB='$BIBT',        SORT LIBRARY
//             OUT='$OUT',             OUTPUT CLASS
//             UWK=$UWK,               WORK UNIT
//             CYL='(3,1)',            SORTWORK SPACE
//             VOLS='SER=$VOLO',       BACKUP VOLUME
//             SPAWK='(TRK,(50,10),RLSE)', SPACE OF WORK FILES
//             SPAMB='(TRK,(5,1),RLSE)',   SPACE OF TRANSACTIONS FILE
//             SPATC='(TRK,(150,10),RLSE)', SPACE OF BACKUP FILE
//             RESLIB='$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=$IRLM
//*--------------------------------------------------------------------*
//COPY    EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&RETAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTA400  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA400,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
//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=&INDEXP..&ROOT.&ROOT.SY(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$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7DR   DD DSN=&&RETAMB,DISP=(OLD,PASS)
//PAC7DE   DD DSN=&&DEREO,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
//            SPACE=&SPAWK
//PAC7TX   DD DSN=&&TXREO,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=VB,LRECL=1063,BLKSIZE=10630),
//            SPACE=&SPAWK
//PAC7IR   DD SYSOUT=&OUT
//*
//PTA410   EXEC PGM=PTA410
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&RETAMB,DISP=(OLD,PASS)
//PAC7TX   DD DSN=&&TXREO,DISP=(OLD,PASS)
//PAC7TW   DD DSN=&&TWREO,DISP=(,PASS),UNIT=&UWK,
//         SPACE=&SPAWK,
//         DCB=(RECFM=VB,LRECL=1063,BLKSIZE=10630)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//*
//PTA420  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA420,PTA420$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.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=PTA430,COND=(8,LE,PTA400)
```

```
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,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=&INDEXQ..&ROOT.&FILE.TC(+1),
//         UNIT=&UWK,VOL=&VOLS,
//         DISP=(NEW,CATLG,DELETE),
//         SPACE=&SPATC,
//         DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PTASVG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTASVG,PTASVG$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),
//         UNIT=&UWK,VOL=&VOLS,
//         DISP=MOD,
//         DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//*
```

21. BACKUP (SVTA)

12.1. INTRODUCTION

TABLE 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

The files must be closed to on-line use.

USER INPUT

None.

12.2. DESCRIPTION OF STEPS

SVTA: DESCRIPTION OF STEPS

```
TD BACKUP:  PTASVD

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
.Output file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(+1)


TV BACKUP:  PTASVV

.Permanent input file:
 -Table-contents file                    : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
.Output file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=MOD


TG BACKUP:  PTASVG

.Permanent input file:
 -User-parameter file
  DSN=&INDEX..&ROOT.&FILE.TG
.Output file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=MOD
```

12.3. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*          - BACKUP OF THE PACTABLES DATABASE (TD, TV, TG) -        *
//*********************************************************************
//$RADP.SVTA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       INDEX OF NON VSAM FILES
//             INDEXQ='$INDEXQ',       GENERATION FILE INDEX
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//             SPATC='(TRK,(30,3),RLSE)', BACKUP-FILE SPACE
//             VOLS='SER=$VOLO',       BACKUP-FILE VOLUME
//             UNITS='$UNITO',         BACKUP-FILE UNIT
//             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=$IRLM
//*--------------------------------------------------------------------*
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//*
//PTASVD  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTASVD,PTASVD$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//*
//PTASVV  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTASVV,PTASVV$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
```

```
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=MOD,
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//*
//PTASVG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTASVG,PTASVG$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=MOD,
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//*
```

13. PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)

13.1. INTRODUCTION

TRANSFER OF A DATABASE FROM ANOTHER PLATFORM

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 converted in the EBCDIC format for IBM-MVS (TCTA procedure).

If the source site version is the same as the target site's version, the actions to perform are the following:

. 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. However, read the notes on the following page carefully.

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 Overrid on the PTATC2.PAC7TC procedure (see the execution test JCL provided).

13.2. DESCRIPTION OF STEPS

TCTA: DESCRIPTION OF STEPS

```
TC BACKUP SPLIT:  PTATC1

.Input backup file
  PAC7TC : (SAVIN parameter of the procedure)

.Output work files:
 -Tables descriptions sequential image
  PAC7SD : DSN=&&PAC7SD
 -Tables contents sequential image
  PAC7SV : DSN=&&PAC7SV
 -Parameter sequential image
  PAC7SG : DSN=&&PAC7SG


TABLE-DESCRIPTION SORT:  PTATCD

.Input work file:
 -Tables descriptions sequential image
  PAC7SD : DSN=&&PAC7SD

.Output work file:
 -Sorted table descriptions
  PAC7AD : DSN=&&PAC7AD


TABLE-CONTENTS SORT:  PTATCV

.Input work file:
 -Sequential image of tables contents
  PAC7SV : DSN=&&PAC7SV

.Output work file:
 -Sorted tables contents
  PAC7AV : DSN=&&PAC7AV


USER-PARAMETER SORT:  PTATCG

.Input work file:
 Sequential image of parameters
  PAC7SG : DSN=&&PAC7SG

.Output work file:
 -Sorted user parameters
  PAC7AG : DSN=&&PAC7AG


RECONSTITUTION OF THE TC BACKUP:  PTATC2

.Permanent output file:
 -TC backup in EBCDIC format
  PAC7TC : DSN=&INDUN..&ROOTT.00TC(+1)

.Input work files:
 -Tables descriptions sequential image
  PAC7AD : DSN=&&PAC7AD
 -Sequential image of contents
  PAC7AV : DSN=&&PAC7AV
 -Sequential image of parameters
  PAC7AG : DSN=&&PAC7AG
```

13.3. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*      ----  TRANSFER OF A PACTABLES DATABASE ---
//*********************************************************************
//$RADP.TCTA PROC ROOT=$ROOT,  PACTABLES SYSTEM ROOT
//         FILE='$FILE',         PHYSICAL BASE NUMBER
//         INDEXQ='$INDEXQ',       BACKUP FILES' INDEX
//*:       VSAMCAT='$CATU',      USER VSAM CATALOG
//         STEPLIB='$MODB',      LOAD-MODULE LIBRARY
//         UWK=$UWK,             WORK UNIT
//         OUT=$OUT,             OUTPUT CLASS
//         SPATC='(TRK,(30,3),RLSE)', BACKUP-FILE SPACE
//         VOLS='SER=$VOLO',       BACKUP-FILE VOLUME
//         UNITS='$UNITO',         BACKUP-FILE UNIT
//         SAVIN='$INDEXQ..$ROOT..$FILE.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=PTATC1
//***********************
//STEPLIB  DD DSN=&STEPLIB,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=PTATCD
//***********************
//STEPLIB  DD DSN=&STEPLIB,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=PTATCV
//***********************
//STEPLIB  DD DSN=&STEPLIB,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=PTATCG
//***********************
//STEPLIB  DD DSN=&STEPLIB,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=PTATC2
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7AD   DD DSN=&&PAC7AD,DISP=SHR
//PAC7AV   DD DSN=&&PAC7AV,DISP=SHR
//PAC7AG   DD DSN=&&PAC7AG,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=MOD,
//         UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//         DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

14. RESTORATION (RSTA)

14.1. INTRODUCTION

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

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.

14.2. DESCRIPTION OF STEPS

RSTA: DESCRIPTION OF STEPS

```
DEFINITION OF FILES:  IDCAMS

.Defined files
 -Table-description file :     PAC7TD
  DSN=&INDEX..&ROOT.&FILE.TD
 -Table-contents file    :     PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
 -User parameter file    :     PAC7TG
  DSN=&INDEX..&ROOT.&FILE.TG


RESTORATION OF TD:  PTARSD

.Permanent output file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Permanent input file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(0)


RESTORATION OF TV:  PTARSV

.Permanent output file:
 -Table contents file         : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU

.Permanent input file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(0)

RESTORATION OF TG:  PTARSG

.Permanent output file:
 -User parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Permanent input file:
 -Table backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(0)
```

14.3. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*            - RELOADING-RESTORING TABLES (TD-TV-TG) -              *
//*********************************************************************
//$RADP.RSTA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//             INDEXQ='$INDEXQ',       GENERATION-FILE INDEX
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//*:          VSAMCAT='$CATU',        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=$IRLM
//*--------------------------------------------------------------------*
//*
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN  DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TD),DISP=SHR
//       DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TU),DISP=SHR
//       DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TV),DISP=SHR
//       DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.TG),DISP=SHR
//*
//PTARSD  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTARSD,PTARSD$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(0),DISP=SHR
//*
//PTARSV  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTARSV,PTARSV$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(0),DISP=SHR
//*
//PTARSG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTARSG,PTARSG$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(0),DISP=SHR
//*
```

15. LIST OF TABLE DESCRIPTIONS (LDTA)

15.1. INTRODUCTION

LDTA: LIST OF TABLE DESCRIPTIONS

INTRODUCTION

This procedure prints descriptions of tables.

EXECUTION CONDITION

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

15.2. USER INPUT

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.

15.3. DESCRIPTION OF STEPS

LDTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


 PRINTING OF TABLES DESCRIPTIONS:  PTA290

.Permanent input file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Input transaction file:
 -Print request
  PAC7TE :DSN=&&LDTAMB

.Output report:
 -Tables descriptions printout
  PAC7ID
```

15.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*               - TABLE-DESCRIPTION LIST -                          *
//*********************************************************************
//$RADP.LDTA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTIONS FILE
//             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=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&LDTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//*
//PTA290  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA290,PTA290$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE   DD DSN=&&LDTAMB,DISP=(OLD,PASS)
//PAC7ID   DD SYSOUT=&OUTL
//*
```

16. PARAMETER UPDATE (PMTA)

16.1. INTRODUCTION

PMTA: USER PARAMETER UPDATE

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 equipment in use allows Batch/TP concurrency.

16.2. USER INPUT

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	ttttt	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
	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 Aministrator code is not authorized.

16.3. DESCRIPTION OF STEPS

PMTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


PARAMETER UPDATE:  PTA100

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE

.Permanent input-output file:
 -User parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Extraction requests
  PAC7MV : DSN=&&PMTAMB

.Output file:
 -Parameter printing requests
  PAC7NU : DSN=&&NUTAB

.Output report:
 -Printing of descriptions
   PAC7ET


PRINTING OF USER PARAMETERS:  PTA120

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -User parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Print requests
  PAC7NU : DSN=&&NUTAB

.Output report:
 -Printing of user parameters
  PAC7ET
```

16.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - UPDATE OF USER PARAMETERS -                      *
//*********************************************************************
//$RADP.PMTA   PROC FILE=$FILE,        PHYSICAL DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//             OUT='$OUT',             OUTPUT CLASS
//             OUTL=$OUT,              OUTPUT CLASS
//             UWK=$UWK,               WORK UNIT
//             SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTION FILES
//             SPANU='(TRK,(5,1),RLSE)',  SPACE OF WORK FILES
//             RESLIB='$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=$IRLM
//*-------------------------------------------------------------------*
//COPY    EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&PMTAMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//*
//PTA100  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA100,PTA100$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.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=$REGSIZ,
//        PARM=(DLI,PTA120,PTA120$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7NU   DD DSN=&&NU,DISP=(OLD,DELETE)
//PAC7ET   DD SYSOUT=&OUTL
//*
```

17. TABLE EXTRACTION (EXTA)

17.1. INTRODUCTION

EXTA: TABLE EXTRACTION

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.

17.2. USER INPUT

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	ttttt	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

17.3. DESCRIPTION OF STEPS

EXTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


EXTRACTION OF TABLE DATA:  PTA150

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -Table-contents file            : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
 -User parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Extraction requests
  PAC7MV : DSN=&&EXTAMB

.Output file:
 -Extracted transactions
  PAC7EX : DSN=&&EX

.Output report:
 -Transaction review
  PAC7ET

PRINTING OF EXTRACTED TRANSACTIONS:  PTA160

.Permanent input file:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD

.Input transaction file:
 -Extracted transactions
  PAC7EX : DSN=&&EX

.Output report:
 -Printing of extracted data
  PAC7ET

.Output file:
 -Extracted transactions
  PAC7NU : DSN=&&MBTAB

.Sort files:
  SORTWK01, SORTWK02, SORTWK03.

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

17.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - EXTRACTION OF TABLES -                           *
//*********************************************************************
//$RADP.EXTA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEX='$INDEX',         VSAM-FILE INDEX
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//*:          SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//             STEPLIB='$MODB',        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 FILES
//             SPAEX='(TRK,(10,10),RLSE)', SPACE OF EXTRACTED TRANSACT.
//             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=PTU001
//STEPLIB  DD DSN=&STEPLIB,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
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTA150  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA150,PTA150$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7MV   DD DSN=&&EXTAMB,DISP=(OLD,PASS)
//PAC7EX   DD DSN=&&EX,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=120,BLKSIZE=2400),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*
//PTA160  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTA160,PTA160$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7EX   DD DSN=&&EX,DISP=(OLD,DELETE)
//PAC7NU   DD DSN=&&MBTAB,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*
```

18. DIRECT CONSULTATION OF TABLES (TUTA)

18.1. INTRODUCTION

TUTA: DIRECT CONSULTATION OF TABLES

INTRODUCTION

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

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.

18.2. USER INPUT

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.

18.3. DESCRIPTION OF STEPS

TUTA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


DIRECT CONSULTATION OF TABLES:  PTAU80

.Permanent input files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.TD
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -Table-contents file                : PAC7TV-PAC7TU
  DSN=&INDEX..&ROOT.&FILE.TV
  DSN=&INDEX..&ROOT.&FILE.TU
 -User-parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Request transactions
  PAC7MX : DSN=&&TUTAMB

.Permanent output files:
 -Tables descriptions file
  DSN=&INDEX..&ROOT.&FILE.AD
 -Tables contents file
  DSN=&INDEX..&ROOT.&FILE.AV

.Output report:
 -Transaction report
  PAC7ET

.Sort files:
  SORTWK01, SORTWK02, SORTWK03.
```

18.4. EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - DIRECT READING OF TABLES -                       *
//*********************************************************************
//$RADP.TUTA PROC FILE=$FILE,   PHYSICAL-DATABASE NUMBER
//           ROOT=$ROOT,        VA PAC SYSTEM ROOT
//           INDEX='$INDEX',    VSAM-FILE INDEX
//           INDEXP='$INDEXP',  INDEX OF NON-VSAM FILES
//*:         SYSTCAT='$CATV',   VSAM SYSTEM CATALOG
//*:         VSAMCAT='$CATU',   VSAM USER CATALOG
//           OUT='$OUT',        OUTPUT CLASS
//           STEPLIB='$MODB',   LIBRARY OF BATCH LOAD-MODULES
//           SORTLIB='$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=$IRLM
//*--------------------------------------------------------------------*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TUTAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.AD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.AV),DISP=SHR
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//DDTV     DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTAU80  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAU80,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
//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=&INDEXP..&ROOT.&ROOT.SY(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$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&INDEX..&ROOT.&FILE.TU,DISP=SHR
//PAC7TV$SUF DD DSN=&INDEX..&ROOT.&FILE.TV,DISP=SHR
//PAC7AD   DD DSN=&INDEX..&ROOT.&FILE.AD,DISP=SHR
//PAC7AV   DD DSN=&INDEX..&ROOT.&FILE.AV,DISP=SHR
//PAC7MX   DD DSN=&&TUTAMB,DISP=(OLD,DELETE)
//PAC7ET   DD SYSOUT=&OUT
//*
```

19. DISPATCHED TABLE MANAGEMENT (DTM OPTION)

19.1. TABLE DESCRIPTION COMPARISON (CDT1-CDT2)

DISPATCHED TABLE MANAGER (DTM)

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 PTAD10 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.

19.2. USER INPUT (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.

19.3. DESCRIPTION OF STEPS (CDT1)

CDT1: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


VERIFICATION OF VSAM FILES:  IDCAMS


CHECK OF TRANSACTIONS:  PTAD05

.Permanent input files:
 -Table description 'Master' file
  PAC7TD : DSN=&TDMAST
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -User-parameter file
  DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
 -Comparison request transactions
  PAC7MV : DSN=&&CDT1MB

.Output file:
 -Validated comparison request transactions
  PAC7MX : DSN=&&MX

.Output report:
 -Transaction report
  PAC7ET


TABLE-DESCRIPTION COMPARISON AND EXTRACTION:  PTAD10

.Permanent input files:
 -Table description 'Master' file
  PAC7TD : DSN=&TDMAST
 -Error message file
  DSN=&INDEX..&ROOT.&ROOT.TE
 -Table description 'Slave' file
  VSAM file
  PAC7TS : DSN=&TDSLAV

.Input transaction file:
 -Validated transactions
  PAC7MX : DSN=&&MX

.Output file:
 -Comparison result to be used as input of
  the CDT2 procedure
  PAC7TX : DSN=&XD

.Output report:
 -Extraction printout
  PAC7ET
```

19.4. EXECUTION JCL (CDT1)

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - TABLE-DESCRIPTION COMPARISON -                   *
//*********************************************************************
//$RADP.CDT1 PROC FILE='$FILE',     PHYSICAL-DATABASE NUMBER
//           ROOT='$ROOT',          VA PAC SYSTEM ROOT
//           INDEX='$INDEX',        VSAM-FILE INDEX
//           INDEXP='$INDEXP',      NON-VSAM FILE INDEX
//*:         VSAMCAT='$CATV',       VSAM USER CATALOG
//*:         SYSTCAT='$CATV',       VSAM SYSTEM CATALOG
//           STEPLIB='$MODB',       LIBRARY OF BATCH PROGRAMS
//           SORTLIB='$BIBT',       SORT LIBRARY
//           TDMAST=,               MASTER DESCRIPTION
//           TDSLAV=,               SLAVE DESCRIPTION
//           XD='&&TX',             DSN OF EXTRACTED DESCRIPTION
//           SPAXD='(TRK,(30,10),RLSE)', SPACE OF EXTRACTED DESCRIPTION
//           SPAMB='(TRK,(5,1),RLSE)',   SPACE OF TRANSACTIONS FILES
//           CYL=3,                 SORTWORK SPACE
//           UWK=$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=PTU001
//STEPLIB  DD DSN=&STEPLIB,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
//DDTD     DD DSN=&TDMAST,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTS     DD DSN=&TDSLAV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTS),DISP=SHR
//*
//PTAD05  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAD05,PTAD05$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
//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=&INDEXP..&ROOT.&ROOT.SY(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$SUF DD DSN=&TDMAST,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7MV   DD DSN=&&CDT1MB,DISP=(OLD,PASS)
//PAC7MX   DD DSN=&&MX,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),
//            SPACE=&SPAXD
//PAC7ET   DD SYSOUT=&OUTL
//*
//PTAD10  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAD10,PTAD10$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&TDMAST,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,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
//*
```

19.5. DESCRIPTION OF STEPS (CDT2)

CDT2: DESCRIPTION OF STEPS

```
VERIFICATION OF VSAM FILES:  IDCAMS


UPDATE OF 'SLAVE' FILES,
RECOGNITION OF THE FILE EXTRACTED BY CDT1: PTAD20

.Input files:
 -Tables descriptions 'Slave' file
  PAC7TD : DSN=&TDSLAV
 -Error-message file
  DSN=&INDEX..&ROOT.&ROOT.TE

.Output file:
 -File of table contents associated to the
  'slave' table-description file             : PAC7TV-PAC7TU
  PAC7TV : DSN=&TVSLAV

.Input transaction file:
 -Result extracted from comparison in the
  CDT1 procedure
  PAC7TX : DSN=&XD

.Output report:
 -Update report
  PAC7ET

.Sort files:
  SORTWK01, SORTWK02, SORTWK03.
```

19.6. EXECUTION JCL (CDT2)

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                  - TABLE-DESCRIPTION UPDATE -                     *
//*********************************************************************
//$RADP.CDT2 PROC ROOT='$ROOT',   SYSTEM ROOT
//           INDEX='$INDEX',      VSAM-FILE INDEX
//           INDEXP='$INDEXP',    NON-VSAM FILE INDEX
//*:         VSAMCAT='$CATV',     VSAM USER CATALOG
//*:         SYSTCAT='$CATV',     VSAM SYSTEM CATALOG
//           STEPLIB='$MODB',     LIBRARY OF BATCH PROGRAMS
//           SORTLIB='$BIBT',     SORT LIBRARY
//           TDSLAV=,             SLAVE DESCRIPTION
//           TUSLAV=,             PRIMARY INDEX 'TU' OF SLAVE DESCR.
//           TVSLAV=,             TABLES ASSOCIATED TO SLAVE DESCRIPT.
//           XD='&&TX',           DSN OF DESCR. EXTRACTED FROM 'CDT1'
//           CYL=3,               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=$IRLM
//*--------------------------------------------------------------------*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&TDSLAV,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTU     DD DSN=&TUSLAV,DISP=SHR
//DDTV     DD DSN=&TVSLAV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTAD20  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAD20,PTAD20$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUF DD DSN=&TDSLAV,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TU$SUF DD DSN=&TUSLAV,DISP=SHR
```

```
//PAC7TV$SUF DD DSN=&TVSLAV,DISP=SHR
//PAC7TX   DD DSN=&XD,DISP=SHR
//PAC7ET   DD SYSOUT=&OUTL
//*
```

19.7. TABLE CONTENTS UPDATE (CVTA)

CVTA: COMPARISON AND UPDATING OF TABLES CONTENTS

INTRODUCTION

The CVTA procedure extracts tables 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.

19.8. USER INPUT (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	ttttt	Table number
9	8	DDMMCCYY	Update date: beginning
17	2		Not used
19	1	'/'	Delimiter
20	1		Not used
21	8	DDMMCCYY	Update date: end

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

19.9. DESCRIPTION OF STEPS (CVTA)

CVTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

TABLE-CONTENTS COMPARISON: PTAV10

```
.Permanent input files:
-Tables descriptions file
 PAC7TD : DSN=&TD
-Error-message file
 DSN=&INDEX..&ROOT.&ROOT.TE
-Table-contents file               : PAC7TV-PAC7TU
 PAC7TV : DSN=&TV
-User-parameter file
 DSN=&INDEX..&ROOT.&FILE.TG

.Input transaction file:
-Comparison requests
 PAC7MV : DSN=&&CVTAMB

.Output file:
-Comparison result
 PAC7EX : DSN=&&EX

.Output report:
 -Transaction report
  PAC7ET
```

EXTRACTION OF UPDATE TRANSACTIONS: PTAV20

```
.Permanent input file:
-Tables Descriptions file
 PAC7TD : DSN=&TD

.Input transaction file:
-Comparison result
 PAC7EX : DSN=&&EX


.Output file:
-Update transactions for use as
 input of UPTA
 PAC7NU : DSN=&&NU

.Output report:
 -Printing of extracted transactions
  PAC7ET

.Sort files:
 SORTWK01, SORTWK02, SORTWK03.
```

19.10. EXECUTION JCL (CVTA)

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*              - EXTRACTION OF MODIFIED TABLE DATA -                *
//*********************************************************************
//$RADP.CVTA PROC FILE='$FILE',    PHYSICAL-DATABASE NUMBER
//           ROOT='$ROOT',         SYSTEM ROOT
//           INDEX='$INDEX',       VSAM-FILE INDEX
//           INDEXP='$INDEXP',     NON-VSAM FILE INDEX
//*:         VSAMCAT='$CATV',      VSAM USER CATALOG
//*:         SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//           STEPLIB='$MODB',      LIBRARY OF BATCH PROGRAMS
//           SORTLIB='$BIBT',      SORT LIBRARY
//           TD=,                  SLAVE DESCRIPTION
//           TU=,                  'TU' PRIMARY INDEX OF SLAVE DESCRIPT
//           TV=,                  TABLES ASSOCIATED TO SLAVE DESCRIPT.
//           SPAEX='(TRK,(30,3),RLSE)', SPACE OF EXTRACTED TRANSACTIONS
//           SPAMB='(TRK,(5,1),RLSE)',  SPACE OF TRANSACTION FILE
//           CYL=3,                SORTWORK SPACE
//           UWK=$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=PTU001
//STEPLIB  DD DSN=&STEPLIB,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
//DDTD     DD DSN=&TD,DISP=SHR
//DDTE     DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//DDTG     DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//DDTU     DD DSN=&TU,DISP=SHR
//DDTV     DD DSN=&TV,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTU),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTV),DISP=SHR
//*
//PTAV10  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAV10,PTAV10$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&TD,DISP=SHR
//PAC7TE$SUF DD DSN=&INDEX..&ROOT.&ROOT.TE,DISP=SHR
//PAC7TG$SUF DD DSN=&INDEX..&ROOT.&FILE.TG,DISP=SHR
//PAC7TU$SUF DD DSN=&TU,DISP=SHR
//PAC7TV$SUF DD DSN=&TV,DISP=SHR
//PAC7MV   DD DSN=&&CVTAMB,DISP=(OLD,PASS)
//PAC7EX   DD DSN=&&EX,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=120,BLKSIZE=2400),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*
//PTAV20  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAV20,PTAV20$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7TD$SUF DD DSN=&TD,DISP=SHR
//PAC7EX   DD DSN=&&EX,DISP=(OLD,PASS)
//PAC7NU   DD DSN=&&NU,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),
//            SPACE=&SPAEX
//PAC7ET   DD SYSOUT=&OUTL
//*
```

20. TABLE RETRIEVAL FROM RELEASES 7.x (RxTA)

20.1. RETRIEVAL FROM RELEASE 7.3, 8.xx or 1.2 (R3TA)

RETRIEVAL FROM RELEASE 7.3, 8.xx or 1.2 (R3TA)

The retrieval of existing tables and files, which allows Tables of the 7.3 release to be used in Pactables 2.5, includes five steps:

. Application of the 2.5 retrieval procedure (R3TA) to the TD, TV, and TG files of Rel. 7.3, producing a backup (TC) formatted to meet the requirements of Rel. 2.5.

Note :

For this procedure, refer to the description of Retrieval 8.02, 1.2 --> 2.5, for details on the definition and use of user input.

. Execution of the 2.5 restoration procedure (RSTA) on the Database, using the backup produced by the preceding step.

. Assignment of the Administrator access authorization level to the Database Manager '********' (PMTA).

. Execution of the 2.5 reorganization procedure (RETA), using the restored Database, in order to purge it, and to assign the sign + to data signed positive (sign missing from release 7.3), for the purpose of user programs written in Cobol II.

. Second execution of the 2.5 restoration procedure (RSTA) on the Database, using the backup produced by the preceding step.

RESULT
Pactables files ready to be used in Release 2.5.

20.2. DESCRIPTION OF STEPS (R3TA)

R3TA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


BACKUP OF THE 7.3 TD FILE:  PTAXVD

This step creates a backup of the TD file used in
release 7.3.

.Input file:
 -7.3 TD file
  PAC7TD : DSN=parameter &OLDTD of the procedure

.Output file:
 -7.3 Backup file
  PAC7TC : DSN=&&R3TATC01


7.3 TV FILE BACKUP:  PTAXVV

This step creates a backup of the TV file of
release 7.3

.Input file:
 -7.3 TV file                    : PAC7TV-PAC7TU
  PAC7TV : DSN=parameter &OLDTV of the procedure

.Output file:
 -7.3 backup file
  PAC7TC : DSN=&&R3TATC01

BACKUP OF 7.3 TG FILE:  PTAXVG

This step creates a backup of the TG file from Release
7.3.

.Input file:
 -7.3 TG file
  PAC7TG : DSN=parameter &OLDTG of the procedure

.Output file:
 -7.3 backup file
  PAC7TC : DSN=&&R3TATC01


CONVERSION OF 7.3 BACKUP INTO 2.5 BACKUP:  PTAR20

This step creates a backup in 2.5 format from the
7.3 backup file.

.Input file:
 -7.3 backup file
  PAC7TC : DSN=&&R3TATC01
 -User-parameter file
  PAC7MB

.Output file:
 -Temporary 2.5 backup
  DSN=&INDEXQ..&ROOT.&FILE.TC(+1)

.Output report:
 -Retrieval report
  PAC7ET
```

20.3. EXECUTION JCL (R3TA)

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*            - RETRIEVAL OF PACTABLES FILES (REL. 7.3) -            *
//*********************************************************************
//$RADP.R3TA   PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//             ROOT=$ROOT,             VA PAC SYSTEM ROOT
//             INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//             INDEXQ='$INDEXQ',       GENERATION-FILE INDEX
//             STEPLIB='$MODB',        LIBRARY OF BATCH PROGRAMS
//*:          VSAMCAT='$CATU',        VSAM USER CATALOG
//             SPATC='(TRK,(30,3),RLSE)', BACKUP-FILE SPACE
//             OLDTD=,                  7.3 TD FILE
//             OLDTG=,                  7.3 TG FILE
//             OLDTV=,                  7.3 TV FILE
//             VOLS='SER=$VOLO',       BACKUP-FILE VOLUME
//             UNITS='$UNITO',         BACKUP UNIT
//             UWK='$UWK',             WORK UNIT
//             OUT='$OUT',             OUTPUT CLASS
//             OUTL='$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
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&R3TAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*
//PTAXVD  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAXVD,PTAXVD$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUFR DD DSN=&OLDTD,DISP=SHR
//PAC7TC   DD DSN=&&R3TATC01,DISP=(,PASS),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=(RECFM=VB,LRECL=1061,BLKSIZE=10614)
//*
//PTAXVV  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAXVV,PTAXVV$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
//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=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TV$SUFR DD DSN=&OLDTV,DISP=SHR
//PAC7TC   DD DSN=*.PTAXVD.PAC7TC,DISP=(MOD,PASS)
//*
//PTAXVG  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAXVG,PTAXVG$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TG$SUFR DD DSN=&OLDTG,DISP=SHR
//PAC7TC   DD DSN=*.PTAXVD.PAC7TC,DISP=(MOD,PASS)
//*
//PTAR20  EXEC PGM=PTAR20
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&R3TAMB,DISP=SHR
//PAC7TC   DD DSN=*.PTAXVD.PAC7TC,DISP=SHR
//PAC7TR   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//PAC7ET   DD SYSOUT=&OUTL
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
```

20.4. RETRIEVAL FROM RELEASE 7.2 (R2TA)

RETRIEVAL FROM RELEASE 7.2 (R2TA)

The retrieval of existing tables and files, which allows Tables of the 7.2 release to be used in the new 2.5 release, includes five steps:

. Conversion of the TG file, Rel. 7.2, into a 7.3 TG file (R2TA procedure); creation of a backup (TA) with this file as well as with the 7.2 TD and TV files, so as to make up a backup in the 7.3 format. This 7.3 backup is then converted into a 2.5 backup.

NOTE :

For this procedure, refer to the description of Retrieval 8.02, 1.2 --> 2.5, for details on the definition and use of user input.

. Execution of the 2.5 restoration procedure (RSTA) on the Database, using the backup produced by the preceding step.

. Assignment of the Administrator access authorization level to the Database Manager '********' (PMTA).

. Execution of the 2.5 reorganization procedure (RETA), using the restored Database, in order to purge it, and to assign the sign + to data signed positive (sign missing from release 7.2), for the purpose of user programs written in Cobol II.

. Second execution of the 2.5 restoration procedure (RSTA) on the Database, using the backup produced by the preceding step.

RESULT

Pactables files ready to be used in Release 2.5.

20.5. DESCRIPTION OF STEPS (R2TA)

R2TA: DESCRIPTION OF STEPS

```
INPUT RECOGNITION: PTU001


CONVERSION OF THE TG FILE FROM 7.2 TO 7.3:  PTARTG

.Input file:
 -Sequential 7.2 TG file
  PAC7AG : DSN=&&TG

.Output file:
 -7.3 TG file
  DSN=&INDEX..&ROOT.&FILE.TG
 -Conversion report
  PAC7ET

.Input-output file:
 -7.2 TD file
  PAC7TD : DSN=parameter &OLDTD of the procedure


BACKUP OF TD FILE, REL. 7.3:  PTAXVD

This step creates a backup of the 7.3 TD file.

.Input file:
 -7.2 TD file
  PAC7TD : DSN=parameter &OLDTD of the procedure

.Output file:
 -7.3 backup file
  PAC7TC : DSN=&&R2TATC01


BACKUP OF TV FILE, REL. 7.3:  PTAXVV

This step creates a backup of the 7.3 TV file.

.Input file:
 -7.2 TV file
  PAC7TV : DSN=parameter &OLDTV of the procedure

.Output file:
 -7.3 backup file
  PAC7TC : DSN=&&R2TATC01


BACKUP OF TG FILE, REL. 7.3:  PTAXVG

This step creates a backup of the 7.3 TG file.

.Input file:
 -7.2 TG file
  DSN=&INDEX..ROOT.&FILE.TG

.Output file:
 -7.3 backup file
  PAC7TC : DSN=&&R2TATC01
```

CONVERSION OF 7.3 BACKUP TO 2.5 BACKUP: PTAR20

This step creates a 2.5 backup from the 7.3 backup.

.Input file:
-7.3 backup file
PAC7TC : DSN=&&R2TATC01
-User-parameter file
PAC7MB

.Output file:
-Temporary 2.5 backup
PAC7TR : DSN=&&R2TATC02

.Output report:
-Conversion report
PAC7ET

CATALOGING OF THE 2.5 BACKUP: IEBGENER

This step catalogs the generation +1 of the 2.5 backup
if no error has been detected during the preceding steps.

.Input file:
-Temporary 2.5 backup
SYSUT1 : DSN=&&R2TATC02

.Output file:
-Cataloged 2.5 backup
DSN=&INDEXQ..&ROOT.&FILE.TC(+1)

20.6. EXECUTION JCL (R2TA)

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*-------------------------------------------------------------------
//* R2TA : NOT AVAILABLE ON IMS PLATFORM
//*-------------------------------------------------------------------
```

21. TABLE RETRIEVAL FROM RELEASES 8.xx OR 1.2

21.1. INTRODUCTION

RETRIEVAL OF PACTABLES 8.xx OR 1.2 (RTTA)

The RTTA procedure retrieves Pactables files from releases 8.xx and 1.2, making it possible to use the resulting files in Pactables Rel. 2.5.

It creates a Pactables database in which all information of the 'DATE' type used for the management of the database's files are converted from the DDMMYY format into the DDMMCCYY format, or from the YYMMDD format into the CCYYMMDD format, so as to integrate the century mark.

Depending on the years present in the dates of the 8.xx or 1.2 release, the century digits are added by reference to a 'pivot' year specified on a parameter line provided as input to the procedure.

The procedure's test job is delivered with the default 'pivot' year '61' (which can be changed). This means that, for a date in Rel. 1.2 whose year is less than '61', the value '20' is assigned to the century. If the year is higher than 61, the value assigned to the century is '19'.

The retrieval procedure processes only those internal dates useful to the management of the Pactables Database files, and not those belonging to user-specific data.

The retrieval of Pactables 8.xx or 1.2 includes the following steps:

. Backup in 8.xx or 1.2 format (SVTA procedure) of the TD, TV, and TG files of Release 8.xx or 1.2;

. Retrieval in 2.5 format (RTTA procedure) of the backup produced by the preceding step, so as to produce a 2.5 backup file;

. Restoration in the 2.5 format (RSTA procedure) of the database, from the backup produced by the preceding step;

. Assignment of the Administrator access level to the Database Manager '********' (PMTA).

. Execution of the 2.5 reorganization procedure (RETA), using the restored Database, in order to purge it, and to assign the sign + to data signed positive (sign missing from release 7.3), for the purpose of user programs written in Cobol II;

. Second execution of the 2.5 restoration procedure (RSTA) on the Database, using the backup produced by the preceding step.

RESULT

Pactables files ready to be used in Release 2.5.

21.2. RTTA : USER INPUT

USER INPUT

. Parameter line defining the 'pivot' year for century assignment.

Pos.	Len.	Value	Meaning
1	2	Numeric other than '00'	Pivot year

21.3. RTTA : DESCRIPTION OF STEPS

RTTA: DESCRIPTION OF STEPS

```
BACKUP RETRIEVAL:  PTAR20

.Permanent input file:
 -1.2 backup file
  DSN=INDEXQ..&ROOT.&FILE.TC(0)

.Permanent output file:
 -Temporary 2.5 backup
  PAC7TR : DSN=&&RTTATC

.Input file:
 -user parameter line
  PAC7MB

.Output report:
 -Retrieval report
  PAC7ET
```

21.4. RTTA : EXECUTION JCL

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                - RETRIEVAL OF PACTABLES 8.02 OR 1.2 -             *
//*********************************************************************
//$RADP.RTTA   PROC FILE='$FILE',      PHYSICAL-DATABASE NUMBER
//             ROOT='$ROOT',           SYSTEM ROOT
//             INDEXQ='$INDEXQ',       GENERATION-FILE INDEX
//             STEPLIB='$MODB',        BATCH-PROGRAM LIBRARY
//             SPATC='(TRK,(30,3),RLSE)', BACKUP-FILE SPACE
//             VOLS='SER=$VOLO',       BACKUP-FILE VOLUME
//             UNITS='$UNITO',         BACKUP-FILE UNIT
//             OUT='$OUT',             OUTPUT CLASS
//             OUTL='$OUT'             OUTPUT CLASS
//*--------------------------------------------------------------------*
//*
//PTAR20  EXEC PGM=PTAR20
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DUMMY
//PAC7TC   DD DSN=&INDEXQ..&ROOT.&FILE.TC(0),DISP=SHR
//PAC7TR   DD DSN=&INDEXQ..&ROOT.&FILE.TC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPATC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.TC
//PAC7ET   DD SYSOUT=&OUTL
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
```

22. COMPATIBILITY BETWEEN PACTABLES 2.5 AND VA PAC 1.6

COMPATIBILITY PACTABLES 2.5 WITH VISUALAGE PACBASE 1.6

If you use Pactables 2.5 and generate table descriptions of VisualAge Pacbase from version 1.6 and higher versions (or from a former version), you need to use the GETA, GETD, and GETI procedures that are supplied with the Pactables 2.5 installation tape, instead of the GETA, GETD, AND GETI procedures supplied with VisualAge Pacbase as they are not compatible with Pactables 2.5. Copy PBGET* procedures into the VA Pac environment and execute PBZZEXEC upon them.

23. INSTALLATION

23.1. INTRODUCTION

INTRODUCTION

The installation procedure includes three phases:

. Preparation of the installation,

. Installation,

. On-line and batch tests.

It uses an installation tape including the release.
The whole installation process is described in this chapter.

Before proceeding with the installation, it is important that the user be familiar with the technical characteristics of the Pactables function described in this manual in order to prepare the environment (disk space, VSAM catalog and space, on-line options, etc.).

You can then start the installation.

PREPARATION

Retrieval of the initial JCL from the tape, and execution of this JCL:

. Backup of installation tape,

. Copy of the complete JCL processing module,

. Retrieval of the Pactables complete installation and operation JCL.

INSTALLATION

See Subchapter 'Installation Process'.

TESTS

. On-line tests,

. Tests of the batch procedures.

23.2. INSTALLATION TAPE

INSTALLATION TAPE

The installation tape (6250 BPI, Standard labels) is the standard Pactables installation tape, containing the following files:

Rank	Label	Lrecl	Block	Contents
01	INST.JCL	80	3,440	Initial prep. JCL
02	INST.MOD		6,144	MM1JCL Load-module (utility of preparation JCL)
03	PACT.JCL	80	3,440	Pactables installation and operations JCL
04	PACT.MBR8		6,144	Batch load-modules
05	PACT.MTR8		6,144	On-line load-modules
06	PAC.TE	90	6,030	Error Messsage file
07	PAC.TCFRA	1067 (VB)	10,674	Backup of User parameter test file (French)
08	PAC.TCENG	1067 (VB)	10,674	Backup of User parameter test file (English)
09	PACT.DBDLIB		6,144	DBD Library
10	PACT.PSBLIB		6,144	PSB Library
11	PACT.MACT		32,000	Macro-Struct. librar. for TUF-TP facility
12	PACT.README	133 (FBA)	5,320	README File Version Infos

23.3. ENVIRONMENT PREPARATION

PHASE 1 : PREPARATION OF THE ENVIRONMENT

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

For the following discussion the symbols yy and zz represent suffixes, and rr a root of your choice. However, the installation can be facilitated by using the following values for the DBD, PSB and Formats:

```
zz = 22      for the batch PSB suffix
yy = 22      for the DBD suffix
rr = P2      for the prefix of the on-line PSB,
                 formats and transactions
```

```
1.  Declaration of the DBD in use:

  DBD names:  PACDTDyy      DATASET names:  PAC7TDyy
              PACDTEyy                      PAC7TEyy
              PACDTGyy                      PAC7TGyy
              PACDTUyy                      PAC7TUyy
              PACDTVyy                      PAC7TVyy
              PACDTZyy                      PAC7TZyy
              PACDT1yy                      PAC7T1yy
              PACDTByy                      PAC7TByy

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

```
2.  Declaration of Pactables transaction codes:
    (Conversational transactions)

a)  APPLCTN  PSB=rrPxxx
    TRANSACT CODE=rrCxxx,MSGTYPE=(SNGLSEG,RESPONSE),
             MODE=SNGL,SEGSIZE=03500,
             SPA=(150,DASD),EDIT=ULC

    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=rrP501
    TRANSACT CODE=tttttttt,MSGTYPE=(SNGLSEG,RESPONSE),
             MODE=SNGL,SEGSIZE=03500,
             SPA=(150,DASD),EDIT=ULC

    With 'tttttttt' : Pactables transaction code.

c)  APPLCTN  PSB=rrP601
    TRANSACT CODE=mmmmmmmm,MSGTYPE=(SNGLSEG,RESPONSE),
             MODE=SNGL,SEGSIZE=03500,
             SPA=(150,DASD),EDIT=ULC

    With 'mmmmmmmm' : transaction code to access
    the Pactables Password Modification transaction.


3.  Declaration of BMPs:

    APPLCTN  PSB=PTA250rr (table generation)
    APPLCTN  PSB=PTA300rr (table update)
```

NOTES:

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

SPAs must not be defined 'FIXED'.

IMS generation parameter sources are found on the tape, in the PDS of the SY system parameters.

23.4. INITIAL JCL

INSTALLATION PREPARATION

It is recommended to copy all the preparation, installation and operation JCLs in a specific PDS.

The first step then consists in the allocation of a PDS file whose characteristics are the following:

- Lrecl=80
- Size: about 60 tracks of a 3380 disk, 30-directory blocks.

(This allocation is not provided and must therefore be determined first by the person in charge of the installation).

Then, copy the initial JCL from the VisualAge Pacbase installation tape ('INST.JCL') into the allocated PDS, using one of the site's utilities (IEBGENER, for example).

INITIAL JCL

The initial JCL contains the JOBs necessary for the generation of Pactables complete installation and operation JCL.

This complete JCL is obtained via a parameterized skeleton JCL ('PACT.JCL'), thus allowing each user's specific needs to be taken into account.

This JCL is processed by an 'INST.MOD' file utility, which generates a JCL taking into account the parameters specified by the user.

The INITIAL JCL includes three JOBs that should be adapted to the site's requirements:

. Copy of the installation tape supplied with the product onto a user medium. This will constitute the Pactables system backup, and it must be used for the actual installation.

. Downloading, via IEBCOPY, of the MM1JCL utility contained in the INST.MOD file into a load-module library already existing on the site or allocated for this purpose. This operation is only executed when installing Pactables for the first time.

. Execution of the MM1JCL which creates the Pactables installation and operation JCL.

Complete the JCL lines for this execution as follows: //STEPLIB DD DSN= <--- library containing the MM1JCL
//SYSUT1=<--- tape or cartridge number

//SYSUT2 DD DSN= <--- recipient file for the complete installation and operations JCL.
This file may be either a member of the PDS initially designed to store all the JCLs, or a sequential file of your choice.

Enter the required parameters (refer to the following subchapters).

KEEP THIS MM1JCL OUTPUT: YOU MAY NEED IT FOR FUTURE REINSTALLATIONS.

```
$CO************ PACTABLES $REL IMS/VS $VV $DATE *****************
=//SEQ
//PACBASE0 JOB (---),'TAPE',CLASS=D,MSGCLASS=A
//ALLOC   EXEC PGM=IEHINITT
//TIBM     DD DISP=SHR,UNIT=(3480,,DEFER),VOL=(,RETAIN,SER=$BDEIBM)
//TINST    DD DISP=SHR,UNIT=(3480,,DEFER),VOL=(,RETAIN,SER=£££££££ ),
//            DCB=DEN=3
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
TINST INITT SER=£££££££,OWNER='£££££££',DISP=REWIND
/*
//PACCOP  PROC INDEX='$INPRO',NAME=XXX,LAB=N
//GENER   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=&INDEX..&NAME,DISP=SHR,
//            VOL=(,RETAIN,REF=*.ALLOC.TIBM),LABEL=&LAB
//SYSUT2   DD DSN=&INDEX..&NAME,DISP=(,KEEP),
//            VOL=(,RETAIN,REF=*.ALLOC.TINST),LABEL=&LAB,
//            DCB=*.SYSUT1
//        PEND
//*
//STEP01  EXEC PACCOP,LAB=01,NAME=JCL,INDEX=INST
//STEP02  EXEC PACCOP,LAB=02,NAME=MOD,INDEX=INST
//STEP03  EXEC PACCOP,LAB=03,NAME=JCL,INDEX=PACT
//STEP04  EXEC PACCOP,LAB=04,NAME=MBR8,INDEX=PACT
//STEP05  EXEC PACCOP,LAB=05,NAME=MTR8,INDEX=PACT
//STEP06  EXEC PACCOP,LAB=06,NAME=TE
//STEP07  EXEC PACCOP,LAB=07,NAME=TCFRA
//STEP08  EXEC PACCOP,LAB=08,NAME=TCENG
//STEP09  EXEC PACCOP,LAB=09,NAME=DBDLIB,INDEX=PACT
//STEP10  EXEC PACCOP,LAB=10,NAME=PSBLIB,INDEX=PACT
//STEP11  EXEC PACCOP,LAB=11,NAME=MACT,INDEX=PACT
//STEP12  EXEC PACCOP,LAB=12,NAME=README,INDEX=PACT
//*
```

```
//PACBASE1 JOB (---),'UTI',CLASS=D,MSGCLASS=A
//*
//COPIE   EXEC  PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,10)
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,10)
//IM       DD DISP=OLD,UNIT=3480,VOL=(,RETAIN,SER=£££££££),
//            DSN=INST.MOD,LABEL=02
//OM       DD DISP=SHR,DSN=£££.£££.£££
//SYSIN    DD *
 C  I=((IM,R)),O=OM
/*
```

```
//PACBASE2 JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//*
//ET010   EXEC PGM=IEFBR14
//DD1      DD DSN=£££££££££££££££,        <-- JOB PDS NAME/NOM
//            UNIT=%%%%,VOL=SER=??????,     <-- UNIT & VOLUME
//            DISP=(,CATLG),
//            SPACE=(TRK,(10,10,10)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//MM1JCL  EXEC PGM=MM1JCL
//STEPLIB  DD DISP=SHR,DSN=£££.£££.£££
//SYSOUT   DD SYSOUT=A
//SYSUT1   DD DSN=PACT.JCL,DISP=OLD,
//            UNIT=3480,VOL=(,RETAIN,SER=£££££££),LABEL=03
//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 DU JCL D'INSTALLATION PAR L'UTILITAIRE 'MM1JCL' --- *
//*      ******************************************************     *
//*                                                                   *
//*      -MODIFIER LA LISTE DES COMMANDES FOURNIES EN PRECISANT LES   *
//*       VARIANTES D'INSTALLATION (S'IL Y EN A), EN DEMANDANT SI     *
//*       NECESSAIRE LA SELECTION DE PORTIONS DE JCL D'INSTALLATION   *
//*       (MODULES DE JCL), EN DONNANT DES VALEURS APPROPRIEES AUX    *
//*       PARAMETRES D'INSTALLATION, ET EN PRECISANT EVENTUELLEMENT   *
//*       DES LIGNES A AJOUTER EN TETE OU EN FIN DE CHAQUE MODULE     *
//*       DE JCL                                                      *
//*-------------------------------------------------------------------*
//*  --- CREATION OF INSTALLATION JCL THROUGH PROGRAM 'MM1JCL' ---     *
//*      ************************************************         *
//*                                                                   *
//*      -MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ENTERING THE    *
//*       INSTALLATION VARIANTS (IF ANY), BY ASKING, IF NECESSARY, A  *
//*       SELECTION OF PARTS OF INSTALLATION JCL (JCL MODULES), BY    *
//*       GIVING THE APPROPRIATE VALUES TO THE INSTALLATION PARAMETERS*
//*       AND BY SPECIYING (OPTIONAL) THE LINES TO BE ADDED AT THE    *
//*       BEGINNING AR AT THE END OF EACH JCL MODULE.                 *
//*********************************************************************
//SYSPRM   DD DUMMY
//SYSUT2   DD DSN=&&PACBASE2,DISP=(,PASS,DELETE),UNIT=£££££,
//            SPACE=(TRK,(10,10),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//SYSIN DD *
===SELL <>                  .LANGUAGE SELECTION (ENG OR FRA)
===SELV SEC                 .PACTABLES SECURITY (RACF ACF2)
                            .PACTABLES SECURITE (RACF ACF2)
===SELV DTM                 .TABLE COMPARAISON UTILITY
                            .GESTIONNAIRE TABLES REPARTIES
===SELV TUF                 .TUF-TP
                            .TUF-TP
===PRM PRFJ=PAC             .JOB NAMES PREFIXES (3 CHARACTERS)
                            .PREFIXE DES NOMS DE JOBS (3 CARACTERES)
===PRM CCPT=<>              .JOB ACCOUNTING CDES (JOB CARDS)
                            .CODE COMPTABLE DES JOBS (CARTES JOB)
===PRM CLASSJ=1             .JOB EXECUTION CLASS (JOB CARDS)
                            .CLASSE EXECUTION DES JOBS (CARTES JOB)
===PRM MSGCL=A              .JCL OUTPUT CLASS (MSGCLASS)
                            .CLASSE DE SORTIE DU JCL (MSGCLASS)
===PRM UTAPE=3480           .CARTRIDGE UNIT
                            .UNITE DE CARTOUCHE
===PRM TAPEI=<>             .NAME OF INSTALLATION CARTRIDGE
                            .NOM DE LA CARTOUCHE D'INSTALLATION
===PRM OUT=A                .JOB SYSOUT CLASS
                            .CLASSE DE SYSOUT DANS LES JOBS
===PRM INDEX='PAC25IMS.IMS' .VSAM FILE INDEXES
                            .INDEX DES FICHIERS VSAM
===PRM INDEXP='PAC25.SEQ'   .NO VSAM FILE INDEXES
                            .INDEX DES FICHIERS NON VSAM
===PRM INDEXQ='PAC25.SEQ'   .GENERATION FILE INDEXES
                            .INDEX DES FICHIERS A GENERATION
===PRM ROOT='P2'            .ROOT OF SYSTEM (2 CHARACTERS)
```

```
                                .RADICAL DU SYSTEME (2 CARACTERES)
    ===PRM HEXA='D7F2'          .ROOT OF SYSTEM (PRM ROOT) IN HEXA
                                .RADICAL DU SYSTEME(PRM ROOT) EN HEXA
    ===PRM FILE='00'            .NUMBER OF USER'S PACTABLES DATABASE
                                .NUMERO DE BASE UTILISATEUR (2 CHIFFRES)
    ===PRM SUF='22'             .SUFFIX OF DBD       (2 CHARACTERS)
                                .SUFFIXE DES DBD     (2 CARACTERES)
    ===PRM SUFR='20'            .SUFFIX OF DBD PREVIOUSLY INSTALLED
                                .SUFFIXE DES DBD PRECEDEMMENT INSTALLES
    ===PRM SUG='22'             .SUFFIX OF BATCH PSB (2 CARACTERES)
                                .SUFF. DES PSB BATCH (2 CARACTERES)
    ===PRM SUFPB='25'           .SUFFIX OF VA PAC DBD (AN-AR-AE)
                                .SUFFIXE DES DBD VA PAC
    ===PRM RADP='T250'          .PREFIX OF CATALOGUED PROCEDURES NAMES
                                .PREFIXE NOMS DE PROCEDURES CATALOGUEES
    ===PRM REGSIZ='1536K'       .SIZE OF THE REGION BATCH PROCEDURES
                                .TAILLE REGION POUR PROCEDURES BATCH
    ===PRM IMSID='CGIB'         .'IMSID' PARAMETER FOR BMP PROCEDURE
                                .PARAMETRE 'IMSID' POUR PROCEDURES BMP
    ===PRM DBRC=N               .USING DBRC IN BATCH PROCEDURES (N=NO)
                                .UTILISATION DBRC DANS PROC.BATCH (N=NO)
    ===PRM IRLM=N               .USING IRLM IN BATCH PROCEDURES (N=NO)
                                .UTILISATION IRLM DANS PROC.BATCH (N=NO)
    ===PRM VOLV=<>              .VOLUME OF SYSTEM VSAM FILES
                                .VOLUME DES FICHIERS SYSTEMES
    ===PRM VOLU=<>              .VOLUME OF USER VSAM FILES
                                .VOLUME DES FICHIERS UTILISATEUR
    ===PRM CATV=<>              .CATALOG OF SYSTEM VSAM FILES
                                .CATALOGUE VSAM FICHIERS SYSTEME
    ===PRM CATU=<>              .CATALOG OF USER VSAM FILES
                                .CATALOGUE VSAM FICHIERS UTILISATEUR
    ===PRM UWK=SYSDA            .WORK UNIT
                                .UNITE DE TRAVAIL
    ===PRM UNITP=3390           .NO VSAM SYSTEM FILES UNIT
                                .UNITE DES FICHIERS SYSTEMES NON VSAM
    ===PRM UNITO=3390           .NO  USER VSAM FILES UNIT
                                .UNITE DES FICHIERS UTILISATEURS NON VSAM
    ===PRM UNITV=3390           .VSAM SYSTEM FILES UNIT
                                .UNITE DES FICHIERS SYSTEMES  VSAM
    ===PRM UNITU=3390           .VSAM USER FILES UNIT
                                .UNITE DES FICHIERS UTILISATEURS VSAM
    ===PRM VOLP=<>              .NO VSAM SYSTEM FILES VOLUME
                                .VOLUME DES FICHIERS SYSTEMES NON VSAM
    ===PRM VOLO=<>              .NO VSAM USER FILES VOLUME
                                .VOLUME FICHIERS UTILISATEUR NON VSAM
    ===PRM MODB='PAC25.PGMLIBBH' .LIBRARY OF BATCH MODULES
                                .BIBLIOTHEQUE DES MODULES BATCH
    ===PRM MODT='PAC25.PGMLIBTP' .LIBRARY OF ON-LINE MODULES
                                .BIBLIOTHEQUE DES MODULES TP
    ===PRM BIBP='SYS1.PROCLIB'  .PROCEDURE LIBRARY
                                .BIBLIOTHEQUE DES PROCEDURES
    ===PRM BIBT='SYS1.SORTLIB'   .SORT LIBRARY
                                 .BIBLIOTHEQUE DE  TRI
    ===PRM MACT='PAC25.MACLIB'   .TUF-TP MACROS LIBRARY
                                 .BIBLIOTHEQUE DES MACROS TUF-TP
    ===PRM PSBLIB='IMSVS.PSBLIB' .PSB LIBRARY
                                 .BIBLIOTHEQUE PSB
    ===PRM DBDLIB='IMSVS.DBDLIB' .DBD LIBRARY
                                 .BIBLIOTHEQUE DBD
    ===PRM ACBLIB='IMSVS.ACBLIB' .ACB LIBRARY
                                 .BIBLIOTHEQUE ACB
    ===PRM RESLIB='IMSVS.RESLIB' .IMS RESLIB
                                 .RESLIB  IMS
    ===PRM PRCLIB='IMSVS.PROCLIB' .IMS PROCLIB
                                 .PROCLIB IMS
    ===PRM TRANT='TABLES'        .TRANSACTION CODE TO PACTABLES
                                 .CODE TRANSACTION CONNEXION A PACTABLES
    ===PRM TRANP='PASSWORD'      .TRANSACTION CODE TO MODIFY THE PASSWORD
                                 .CODE TRANS. MODIFICATION MOT DE PASE
    ===PRM CLS='2'               .CLASS FOR TRANSACTION CODES
                                 .CLASSE POUR LES CODES TRANSACTION
    ===BEGMOD
    ./ ADD NAME=$MODULE
```

```
/*
//*
//ET020   EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$S
//SYSUT1   DD DSN=*.ET010.DD1,DISP=SHR
//SYSUT2   DD DSN=*.ET010.DD1,DISP=SHR
//SYSIN    DD DSN=&&PACBASE2,DISP=(OLD,DELETE)
//
```

23.5. COMPLETE JCL INSTALLATION

COMPLETE JCL INSTALLATION

The MM1JCL module reads the JCL skeleton file (label 03) and outputs a complete JCL. It allows you to:

. Select the installation language (French or English),

. Select portions of the skeleton JCL, which are called 'JCL modules',

. Parameterize the skeleton in order to obtain a JCL requiring a minimum of modifications to make it operational,

. Select the installation variants to generate the JCL needed for specific processing, depending on site and installation conditions,

. 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 MM1JCL commands
.Table of variants
.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 ($VCAT) */
                    or:       /*: CATALOG ($SCAT) */
-In JCL STEPCATs  as:        //*: STEPCAT  DD
             and/or:         //*:          DD
-In procedure parameters as: //*: VSAMCAT='$VCAT'
                          or: //*: SYSTCAT='$SCAT'
```

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

When these parameters are required, affected lines should be changed into command lines. This is accomplished by:

- Transforming all '//*:' into '//',

- Substituting blanks for '/*:' and '*/'.

Blocking criteria for large files can also be changed.

CAUTION: SMS

. If the SMS product is installed, you should delete IDCAMS definition DD //GDGMOD lines in the installation JCLs with DataGroup Generation allocation.

. If the UNIT and VOL parameters cannot be used on the site, you can delete them in the whole JCL through an exclusion (EXCLUDE command under TSO/EDIT).

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

2. JCL splitting

In front of 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).

This allows for the complete JCL to be split up into as many members as there are JCL modules in a PDS. The completed JCL file is to be used as SYSIN for the PDS update utility: IEUBUPDTE.

NOTE :

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

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

PRINTED OUTPUT

MM1JCL produces a list for each JCL module created, including parameters taken into account and according to required variants.

NOTE :

Since the JCL skeleton parameters are in the $xxxx format, during execution, if MM1JCL encounters a $ character that 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'.
Those messages do not stop the execution and should be ignored: they apply to the '$' in the flow processed by MM1JCL which are NOT parameters.

CODING OF MM1JCL COMMANDS

```
===SELL lll               .Selection of the installation
                           language

===SELV vvvv              .Selection of variant
                           vvvv = variant code

===SELM mmmm1 mmmm2 ...   .Selection of JCL modules
                           mmmm1 = name of JCL module
                           mmmm2 = name of JCL module
                           etc.
                           The absence of a ===SELM line
                           involves the selection of all
                           JCL modules.

===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 jut out
      over 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         )
```

23.6. INSTALLATION DEFAULT SETTINGS

INSTALLATION DEFAULT SETTINGS

.VARIANTS (===SELV): all available variants are selected.

IMPORTANT: DELETE THE LINES CORRESPONDING TO THE VARIANTS NOT INSTALLED ON THE SITE OR INCOMPATIBLE.

.PARAMETERS (===PRM):

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

.MODULES (===SELM):

No selection; all modules (corresponding to the variants) are copied.

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

A line: ./ ADD NAME=$MODULE

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

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

23.7. JCL VARIANTS

TABLE OF VARIANTS

===SELV VVVV .Comments

VVVV	MEANING	TYPE
SEC	Security Systems Interface	Extension
DTM	Dispatched Table Manager	''
TUF	TUF-TP Facility	''

Read the Pactables JCL file:
DSN=PACT.JCL,LABEL=(03,SL), using the MM1JCL module in order to create an adjusted JCL.

TABLE OF JCL MODULES: ===SELM mmmm1 mmmm2 ... mmmmn

mmmm	Contents	Nature
PACTSY	IEBUPDTE : Load PDS parameters	OS JCL
PACTPROC	IEBUPDTE : Load Batch procedures	OS JCL
PACTDBD	IEBCOPY : Load DBD load-modules	OS JCL
PACTPSB	IEBCOPY : Load PSB load-modules	OS JCL
PACTSYS	IEBCOPY : Load Load-modules	OS JCL
PACTTE	Loading of TE user database	OS JCL
PACTTC	Loading of test backup	OS JCL
PACTTZ	Loading of SPA databases (TZ)	OS JCL
PACTTB	Loading of TUF-TP databases (TB)	OS JCL
PACTTAB	Loading of TG TD TV test databases	OS JCL
PACTACB	ACBGEN : generation of ACBs	OS JCL
PACTCTF	download TUF-TP macro-structures	OS JCL
TO2R73	Retrieval 7.3 ---> 2.0	OS JCL
TO2TTA	Retrieval 8.02 or 1.2 ---> 2.0	OS JCL
TO2CD1	Test CDT1 procedure	OS JCL
TO2CD2	Test CDT2 procedure	OS JCL
TO2CVT	Test CVTA procedure	OS JCL
TO2EXT	Test EXTA procedure	OS JCL
TO2GET	Test GETT procedure	OS JCL
TO2IMT	Test IMTA procedure	OS JCL
TO2INT	Test INTA procedure	OS JCL
TO2LPT	Test LPTA procedure	OS JCL
TO2PMT	Test PMTA procedure	OS JCL
TO2PRT	Test PRTA procedure	OS JCL
TO2RET	Test RETA procedure	OS JCL
TO2RST	Test RSTA procedure	OS JCL
TO2SVT	Test SVTA procedure	OS JCL
TO2TCT	Test TCTA procedure	OS JCL
TO2TUT	Test TUTA procedure	OS JCL
TO2UPT	Test UPTA procedure	OS JCL

23.8. JCL PARAMETERIZATION

PARAMETER CHART

```
===PRM PPPP=pppp                .Comments

+-------+-------------------------------------+------------+
!CODE   ! MEANING                             !  DEFAULT   !
!PPPP   !                                     !  pppp      !
+-------+-------------------------------------+------------+
!       !        ON THE JOB CARDS             !            !
!       !        ----------------             !            !
!PRFJ   ! job name prefix                      !  PAC       !
!CCPT   ! Job accounting information           !  <>        !
!CLASSJ ! Job execution class                 !  1         !
!MSGCL  ! Output class                        !  A         !
!       !                                     !            !
!       !       CODING OF FILE DSNAMES        !            !
!       !-------------------------------------!            !
!       !The names of all permanent Pactables !            !
!       !files, except load-module libraries, !            !
!       !are coded as follows:                !            !
!       !-------------------------------------!            !
!       ! INDEX.xxxxss  ! VSAM System Files   !            !
!       ! INDEXP.xxxxss ! NON VSAM System     !            !
!       ! INDEX.xxnnss  ! VSAM User Files     !            !
!       ! INDEXQ.xxnnss ! NON VSAM User Files !            !
!       !                         (GDG)       !            !
!       !                                     !            !
!       ! IND--  Index of file names:         !            !
!INDEX  ! VSAM-system and user                 !PAC25IMS.IMS!
!INDEXP ! NON-VSAM-system (SAM, PDS)           !PAC25.SEQ   !
!INDEXQ ! NON-VSAM-user (GDG)                  !PAC25.SEQ   !
!       !                                     !            !
!       ! xx=ROOT, nn=file ss=file code       !            !
!ROOT   ! Root of the Pactables System        ! P2         !
!       ! (2 characters other than 'ZZ')      !            !
!FILE   ! Pactables database user number      ! 00         !
!       !(2 numeric charact. other than '99') !            !
+-------+-------------------------------------+------------+
```

PARAMETER CHART (continued)

CODE PPPP	MEANING	DEFAULT pppp
	ON THE DD CARDS	
OUT	SYSOUT class	A
UTAPE	Unit type	TAPE
TAPEI	Serial number of your copy of the installation tape provided by IBM.	<>
UWK	Unit of user workfile	SYSDA
UNITP	Unit of NON VSAM System files	SYSDA
UNITO	Unit of NON VSAM User files	SYSDA
UNITV	Unit of VSAM System files	SYSDA
UNITU	Unit of VSAM User files	SYSDA
VOLP	NON VSAM System Files Volume Name	<>
VOLV	VSAM System Files Volume Name	<>
VOLO	NON VSAM User Files Volume Name	<>
VOLU	VSAM User Files Volume Name	<>
	OTHER PARAMETERS	
HEXA	Hexadecimal value of the system ROOT (see parameter ROOT)	D7F2
RADP	Prefix of procedures to install (4 characters maximum) Do not modify this parameter unless T250, already used for Pactables procedures, is not suitable	T250
REGSIZ	Region size for batch procedures	1536K
IMSID	IMSID parameter for BMP procedures	CGIB
DBRC	DBRC use in batch procedures	N
IRLM	IRLM use in batch procedures	N

CODE PPPP	MEANING	DEFAULT pppp
CATU	VSAM Catalog DSNAME on which the Test database set up at installation is to be cataloged	<>
CATV	VSAM Catalog dsname for the Table Management files.	<>
MODB	DSNAME of batch load-modules Library for Pactables	PAC25. PGMLIBBH
MODT	DSNAME of on-line load-modules Library for Pactables	PAC25. PGMLIBTP
BIBP	DSNAME of Library for cataloged Pactables procedures	SYS1.PROCLIB
BIBT	User Sort Library DSNAME	SYS1.SORTLIB
MACT	TUF-TP Macro-structures library	PAC25.MACLIB
PSBLIB	PSB Library DSNAME	IMSVS.PSBLIB
DBDLIB	DBD Library DSNAME	IMSVS.DBDLIB
RESLIB	IMS RESCLIB Library DSNAME	IMSVS.RESLIB
PRCLIB	IMS PROCLIB Library DSNAME	IMSVS.PROCLI
ACBLIB	IMS ACBLIB Library DSNAME	IMSVS.ACBLIB
SUF	DBD suffix	22
SUFR	DBD suffix for previously installed release (different from SUF)	73
SUFPB	VA PAC DBD suffix (AN, AR, AE)	22
SUG	Batch PSB suffix	22
TRANT	Pactables connection transct. code	TABLES
TRANP	Transaction code for Modification of User Passwords	PASSWORD
CLS	TP transaction class	2

Note: The '<>' indicates a parameter which must be coded.

Parameter values containing special characters must be delimited by quotes.

23.9. JCL MODULE SEPARATORS

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 MM1JCL 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 MM1JCL 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.

23.10. INSTALLATION PROCESS

PHASE 3 : INSTALLATION

The JCL obtained contains 12 installation jobs:

```
 1.$prfj.TSY   : Loading the parameters in PDS
 2.$prfj.TPROC : Loading the Batch procedures
 3.$prfj.TDBD  : Loading DBDLIB (Load-modules)
 4.$prfj.TPSB  : Loading PSBLIB (Load-modules)
 5.$prfj.TSYS  : Loading programs, system database/files
 6.$prfj.TTE   : Initial loading of TE
 7.$prfj.TTC   : Initial loading of TC (TG TD TV backup)
 8.$prfj.TTZ   : Loading of the TZ database (SPA)
 9.$prfj.TTB   : Loading of the TB database (TUF-TP)
10.$prfj.TTAB  : Loading of TG, TV and TD test databases
11.$prfj.TACB  : Compiling of the ACBs
12.$prfj.TCTF  : Loading of TUF-TP macrostructures.
```

The installation is carried out in the order of the obtained JOBs, some of which are optional.

1. LOADING THE SYSTEM PARAMETERS

```
ET010: IEBUPDTE: Loading the 'parameters' which represent:
       - Definitions (DELETE/DEFINE) and Verifications
         (VERIFY) of VSAM files.
       - The DBD and PSB source files
       - The DFSVSAMP definition
       - The PACTCTRL member used by the System staff:
         It contains the descriptions of macro-instructions
         'APPLCTN' and 'TRANSACT' of the Pactables function.
         Both macros must be defined in the IMS CONTROL RE-
         GION.
```

2. LOADING THE BATCH PROCEDURES

```
ET010: IEBUPDTE: Loading the operation procedures in a
       PROCLIB.  Each procedure is a member coded as
       '$RADP.NNNN', where $RADP is the root chosen at the
       generation of the JCL and NNNN, the standard
       procedure name. Each procedure is described in a spe-
       cific chapter in this manual.
```

3. LOADING THE DBDLIB

```
The installation tape contains the DBDs of the databases as
object and source code.
The $PRFJ.TDBD Job executes the loading of the DBD object
code into the DBDLIB via IEBCOPY.
The loading is to be executed only if the
values selected in the Generation JCL are as follows:

For:                                        Value:
 A) The DBD Suffix ('SUF' parameter).......:  '22'
 B) The disk type supporting the database
      ('UNITU' and 'UNITV' parameters).....:  3390
And
 C) The C.I. size provided in the parameters of the
    DEFINE of the databases has not been altered.
```

Otherwise, a recompiling of the source must be executed. It would then be necessary to verify the contents and specifically, if the C.I. size has been changed, to adjust the size of the macros.

The source has been cataloged in the Parameter File (SY) under the name PACDxxyy, with yy = value of 'SUF' parameter, and xx = TD, TE, TG, TU, TV, TZ, T1 and TB.

The Compile JCL is not provided.

Space used: Refer to the summary chart in subchapter 'SYSTEM SIZE REQUIREMENT, chapter ENVIRONMENT, in the VisualAge Pacbase ENVIRONMENT AND INSTALLATION Manual.

4. LOADING OF THE PSBLIB

The installation tape contains the PSBs as object and source code.

The $PRFJ.TPSB Job executes the loading of the PSB object code into the PSBLIB via IEBCOPY.

The loading is to be executed only if the values selected in the Generation JCL are as follows:

```
For:                                        Value:
 A) The DBD Suffix ('SUF' parameter).......:  '12'
 B) The root of the system ('ROOT' param.).:  'P2'
 C) The batch PSB suffix ('SUG' parameter).:  '22'
 D) The suffix of previously installed DBSs:  '73'
    (Parameter SUFR used in PTAXVG, PTAXVV, PTAXVD)
    (7.3 Retrieval procedure)
```

Otherwise, a recompiling of the source must be executed. The sources are catalogued in the Parameters File (SY), under the name xxxxxxzz for batch PSBs, rryyyy for on-line PSBs, with zz = value of the 'SUG' parameter, rr = value of 'ROOT' parameter,

```
    xxxxxx= PTABVS PTAINI PTASTD
     PTAT10 PTAT15 PTAT17 PTAU80 PTAXDT PTAXVD PTAXVV PTAXVG
     PTA100 PTA120 PTA150 PTA160 PTA250 PTA290 PTA300 PTA302
     PTA310 PTA312 PTA320 PTA350 PTA360 PTA400 PTA420 PTARSD
     PTARSG PTARSV PTASVV PTASVD PTASVG PTAT19 PTAT40 PTAREO

for the DISPATCHED TABLE MANAGEMENT function:
PTAD05 PTAD10 PTAD20 PTAV10 PTAV20

and for yyyy = P500 P501 P510 P512 P520 P522 P530 P540 P550
               P560 P570 P580 P590 P599 P600 P601 PLNK
and for the DISPATCHED TABLE MANAGEMENT function: P610 P620

The Compile JCL is not provided.

Space used:  Refer to the summary chart in Subchapter
'SYSTEM SIZE REQUIREMENT', chapter ENVIRONMENT, in the
VisualAge Pacbase ENVIRONMENT AND INSTALLATION Manual.
```

5. LOADING OF LOAD-MODULES, SYSTEM DATABASES AND FILES

ET010: IEBCOPY: Loading the on-line and batch load-modules.

ET015: IMASPZAP:

ET016: IMASPZAP: to be executed only if the root supplied, 'P2, is changed to another value. The selected value must be coded in hexadecimal for the ZAPs of $ROOT. P501 and $ROOT.P601.

ET017: IEBCOPY (Security System Interface Extension - SEC option only). This step allows to load the PACSECU sub-program in an authorized program library.

6. LOADING OF ERROR MESSAGES DATABASE (TE)

ET010: IEHPROGM: SCRATCH-UNCATLG of the file whose suffix is IN (source of the error messages).

ET020: IDCAMS : Copy the error messages onto the file whose suffix is IN (source of the errror messages).

ET030: $RADP.LDTE : Loading of TE error message database.

7. LOADING OF THE SPA DATABASE (TZ)

ET010: $RADP.LDTZ: Loading the SPA database (TZ).

This job is only run during the first installation.

8. LOADING OF THE TUF-TP WORKING DATABASE (TB)

ET010 : $RADP.LDTB : Loading of the TUF-TP database.

This job is run only during the first installation.

9. CREATING THE GENERATION BACKUP FILE (TC)

ET010: IEHPROGM: Creating a model DSCB

ET015: IEFBR14

ET020: IDCAMS: Creating a file

ET040: IEBGENER: Loading TC(0) from the installation tape test database

10. LOADING OF TEST DATABASES (TD, TV and TG)

ET010: $radp.RMTA : loading TD, TV and TG test databases from TC files

11. COMPILING ACBs

ET010: ACBGEN.

This step constructs all the ACB's needed for Pactables on-line use.

It must be executed after the loading of the DBDLIB and PSBLIB (either by copying the object modules or by the compiling of the DBDs and the PSBs).

12. VA PAC MACRO-STRUCTURES FOR THE USE OF TUF FACILITY

JOB $prfj.TCTF

This facility loads VA Pac Macro-structures required for the use of the TUF-TP facility.

These Macros-structures are supplied in the form of transactions which are to be used as input to the UPDT procedure. It is then required to update the VA Pac Database before using the TUF-TP facility and this, in order to take these Macro-structures into account.

STEP1: IDCAMS: DELETE Macro-structures library
STEP2: IDCAMS: Allocation of Macro-structures library
STEP3: IEBCOPY: Loading of Macro-structures in the library already allocated.

USER OPERATION COMPLEMENTS

The backups and restorations of Pactables databases and files must be made by utilities.

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 file.

23.11. JCL: LOADING OF PARAMETERS IN 'SY'

```
//$PRFJ.TSY JOB ($CCPT),'LOAD TABLE PARAM.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*************************************************************
//*      LOADING PACTABLES PARAMETERS IN THE 'SY' 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.                    *
//*    *-------------------------------------------------*
//*
//ET005   EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=$OUT
//DD1      DD UNIT=$UNITP,VOL=SER=$VOLP,DISP=SHR
//SYSIN    DD *
    UNCATLG DSNAME=$INDEXP..$ROOT.$ROOT.SY
    SCRATCH DSNAME=$INDEXP..$ROOT.$ROOT.SY,VOL=$UNITP=$VOLP
//*
//ET007   EXEC PGM=IEFBR14
//DDA      DD DSN=$INDEXP..$ROOT.$ROOT.SY,DISP=(,CATLG,DELETE),
//            UNIT=$UNITP,VOL=SER=$VOLP,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(03,02,05))
//*
//ET010   EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2 DD DSN=$INDEXP..$ROOT.$ROOT.SY,DISP=SHR
//SYSIN  DD *
:/ ADD NAME=PACDTD$SUF
       DBD     NAME=PACDTD$SUF,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TD$SUF,DEVICE=$UNITU,                  $A
               RECORD=248,SIZE=4096
       SEGM    NAME=PAC7TD,BYTES=240
       FIELD   NAME=(CLETD,SEQ,U),BYTES=21,START=1
       DBDGEN
       END
:/ ADD NAME=PACDTE$SUF
       DBD     NAME=PACDTE$SUF,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TE$SUF,DEVICE=$UNITU,                  $A
               RECORD=98,SIZE=4096
       SEGM    NAME=PAC7TE,BYTES=90
       FIELD   NAME=(CLETE,SEQ,U),BYTES=17,START=1
       DBDGEN
       END
:/ ADD NAME=PACDTG$SUF
       DBD     NAME=PACDTG$SUF,ACCESS=(HISAM,VSAM)
       DATASET DD1=PAC7TG$SUF,DEVICE=$UNITU,                  $A
               RECORD=94,SIZE=4096
       SEGM    NAME=PAC7TG,BYTES=85
       FIELD   NAME=(CLETG,SEQ,U),BYTES=22,START=1
       DBDGEN
       END
:/ ADD NAME=PACDTU$SUF
       DBD     NAME=PACDTU$SUF,ACCESS=(INDEX,VSAM)
       DATASET DD1=PAC7TU$SUF,DEVICE=$UNITU
       SEGM    NAME=IPAC7TU,BYTES=35
       FIELD   NAME=(CLETU,SEQ,U),BYTES=35,START=1
```

```
        LCHILD  NAME=(PAC7TV,PACDTV$SUF),INDEX=CLETV
        DBDGEN
        END
  :/ ADD NAME=PACDTV$SUF
        DBD     NAME=PACDTV$SUF,ACCESS=(HIDAM,VSAM)
        DATASET DD1=PAC7TV$SUF,DEVICE=$UNITU,                  $A
                SIZE=4096
        SEGM    NAME=PAC7TV,BYTES=(1100,0080)
        FIELD   NAME=(CLETV,SEQ,U),BYTES=35,START=5
        LCHILD  NAME=(IPAC7TU,PACDTU$SUF),PTR=INDX
        DBDGEN
        END
  :/ ADD NAME=PACDTZ$SUF
        DBD     NAME=PACDTZ$SUF,ACCESS=(HISAM,VSAM)
        DATASET DD1=PAC7TZ$SUF,DEVICE=$UNITU,                  $A
                RECORD=8510,SIZE=10240
        SEGM    NAME=PAC7TZ,BYTES=8500
        FIELD   NAME=(CLETZ,SEQ,U),BYTES=11,START=1
        DBDGEN
        END
  :/ ADD NAME=PACDT1$SUF
        DBD     NAME=PACDT1$SUF,ACCESS=(INDEX,VSAM)
        DATASET DD1=PAC7T1$SUF,DEVICE=$UNITU
        SEGM    NAME=IPAC7T1,BYTES=63
        FIELD   NAME=(CLET1,SEQ,U),BYTES=63,START=1
        LCHILD  NAME=(PAC7TB,PACDTB$SUF),INDEX=CLETB
        DBDGEN
        END
  :/ ADD NAME=PACDTB$SUF
        DBD     NAME=PACDTB$SUF,ACCESS=(HIDAM,VSAM)
        DATASET DD1=PAC7TB$SUF,DEVICE=$UNITU,                  $A
                SIZE=4096
        SEGM    NAME=PAC7TB,BYTES=(1140,0080)
        FIELD   NAME=(CLETB,SEQ,U),BYTES=63,START=4
        LCHILD  NAME=(IPAC7T1,PACDT1$SUF),PTR=INDX
        DBDGEN
        END
  :/ ADD NAME=PTABVS$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PSBGEN  PSBNAME=PTABVS$SUG,LANG=COBOL,CMPAT=YES
        END
 ===SEQ FOR DTM
  :/ ADD NAME=PTAD05$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,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$SUF,PROCOPT=A,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PSBGEN  PSBNAME=PTAD20$SUG,LANG=COBOL,CMPAT=YES
        END
 ===SEQ
  :/ ADD NAME=PTAINI$SUG
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=L,KEYLEN=21
```

```
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=L,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PSBGEN  PSBNAME=PTAINI$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAREO$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=L,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=L,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PSBGEN  PSBNAME=PTAREO$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTARSG$SUG
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=L,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PSBGEN  PSBNAME=PTARSV$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTARSD$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=L,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PSBGEN  PSBNAME=PTARSD$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTASTD$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PSBGEN  PSBNAME=PTASTD$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTASVG$SUG
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=GE,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PSBGEN  PSBNAME=PTASVV$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTASVD$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GE,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PSBGEN  PSBNAME=PTASVD$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAT10$SUG
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=L,KEYLEN=17
          SENSEG    NAME=PAC7TE
          PSBGEN  PSBNAME=PTAT10$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAT15$SUG
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=L,KEYLEN=11
          SENSEG    NAME=PAC7TZ
          PSBGEN  PSBNAME=PTAT17$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAT19$SUG
          PCB     TYPE=DB,DBDNAME=PACDTB$SUF,PROCOPT=L,KEYLEN=63
          SENSEG    NAME=PAC7TB
          PSBGEN  PSBNAME=PTAT19$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAT40$SUG
          PCB     TYPE=DB,DBDNAME=PACDAN$SUFPB,PROCOPT=GOT,KEYLEN=43
          SENSEG    NAME=PAC7AN
          PCB     TYPE=DB,DBDNAME=PACDAR$SUFPB,PROCOPT=GOT,KEYLEN=07
          SENSEG    NAME=PAC7AR
          PCB     TYPE=DB,DBDNAME=PACDAE$SUFPB,PROCOPT=GOT,KEYLEN=12
```

```
          SENSEG    NAME=PAC7AE
          PSBGEN  PSBNAME=PTAT40$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAU80$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG  NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
          SENSEG  NAME=PAC7TE
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
          SENSEG  NAME=PAC7TV
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
          SENSEG  NAME=PAC7TG
          PSBGEN  PSBNAME=PTAU80$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAXVG$SUG
          PCB     TYPE=DB,DBDNAME=PACDTG$SUFR,PROCOPT=GE,KEYLEN=22
          SENSEG    NAME=PAC7TG
          PSBGEN  PSBNAME=PTAXVG$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAXVV$SUG
          PCB     TYPE=DB,DBDNAME=PACDTV$SUFR,PROCOPT=GE,KEYLEN=33
          SENSEG    NAME=PAC7TV
          PSBGEN  PSBNAME=PTAXVV$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAXVD$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUFR,PROCOPT=GE,KEYLEN=19
          SENSEG    NAME=PAC7TD
          PSBGEN  PSBNAME=PTAXVD$SUG,LANG=COBOL,CMPAT=YES
          END
   ===SEQ FOR DTM
   :/ ADD NAME=PTAV10$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG  NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
          SENSEG  NAME=PAC7TE
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
          SENSEG  NAME=PAC7TV
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
          SENSEG  NAME=PAC7TG
          PSBGEN  PSBNAME=PTAV10$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTAV20$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG  NAME=PAC7TD
          PSBGEN  PSBNAME=PTAV20$SUG,LANG=COBOL,CMPAT=YES
          END
   ===SEQ
   :/ ADD NAME=PTAXDT$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG  NAME=PAC7TD
          PSBGEN  PSBNAME=PTAXDT$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTA100$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
          SENSEG    NAME=PAC7TE
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
          SENSEG    NAME=PAC7TE
```

```
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PSBGEN  PSBNAME=PTA160$SUG,LANG=COBOL,CMPAT=YES
         END
   :/ ADD NAME=PTA250$SUG
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=A,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PSBGEN  PSBNAME=PTA250$SUG,LANG=COBOL,CMPAT=YES
         END
   :/ ADD NAME=PTA290$SUG
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PSBGEN  PSBNAME=PTA300$SUG,LANG=COBOL,CMPAT=YES
         END
   :/ ADD NAME=PTA302$SUG
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PSBGEN  PSBNAME=PTA302$SUG,LANG=COBOL,CMPAT=YES
         END
   :/ ADD NAME=PTA310$SUG
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PSBGEN  PSBNAME=PTA310$SUG,LANG=COBOL,CMPAT=YES
         END
   :/ ADD NAME=PTA312$SUG
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PSBGEN  PSBNAME=PTA312$SUG,LANG=COBOL,CMPAT=YES
         END
   :/ ADD NAME=PTA320$SUG
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
```

```
          SENSEG    NAME=PAC7TV
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=GOT,KEYLEN=22
          SENSEG    NAME=PAC7TG
          PSBGEN  PSBNAME=PTA320$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTA350$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PSBGEN  PSBNAME=PTA350$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=PTA360$SUG
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=GOT,KEYLEN=17
          SENSEG    NAME=PAC7TE
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=GOT,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,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$SUF,PROCOPT=GOT,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PSBGEN  PSBNAME=PTA420$SUG,LANG=COBOL,CMPAT=YES
          END
   :/ ADD NAME=$ROOT.PLNK
          PCB  TYPE=TP,MODIFY=YES
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=G,KEYLEN=22
          SENSEG    NAME=PAC7TG
          PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
          SENSEG    NAME=PAC7TZ
          PSBGEN  PSBNAME=$ROOT.PLNK,LANG=COBOL
          END
   :/ ADD NAME=$ROOT.P500
          PCB  TYPE=TP,MODIFY=YES
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=A,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
          SENSEG    NAME=PAC7TE
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=G,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=A,KEYLEN=22
          SENSEG    NAME=PAC7TG
          PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
          SENSEG    NAME=PAC7TZ
          PSBGEN  PSBNAME=$ROOT.P500,LANG=COBOL
          END
   :/ ADD NAME=$ROOT.P501
          PCB  TYPE=TP,MODIFY=YES
          PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
          SENSEG    NAME=PAC7TZ
          PSBGEN  PSBNAME=$ROOT.P501,LANG=COBOL
          END
   :/ ADD NAME=$ROOT.P510
          PCB  TYPE=TP,MODIFY=YES
          PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
          SENSEG    NAME=PAC7TD
          PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
          SENSEG    NAME=PAC7TE
          PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
          SENSEG    NAME=PAC7TV
          PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=G,KEYLEN=22
          SENSEG    NAME=PAC7TG
```

```
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P510,LANG=COBOL
         END
   :/ ADD NAME=$ROOT.P512
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=G,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P512,LANG=COBOL
         END
   :/ ADD NAME=$ROOT.P520
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=G,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P520,LANG=COBOL
         END
   :/ ADD NAME=$ROOT.P522
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=A,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=G,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P522,LANG=COBOL
         END
   :/ ADD NAME=$ROOT.P530
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=G,KEYLEN=35
         SENSEG    NAME=PAC7TV
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P530,LANG=COBOL
         END
   :/ ADD NAME=$ROOT.P540
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P540,LANG=COBOL
         END
   :/ ADD NAME=$ROOT.P550
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
```

```
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P550,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P560
        PCB  TYPE=TP,MODIFY=YES
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=G,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P560,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P570
        PCB  TYPE=TP,MODIFY=YES
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P570,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P580
        PCB  TYPE=TP,MODIFY=YES
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTV$SUF,PROCOPT=G,KEYLEN=35
        SENSEG    NAME=PAC7TV
        PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P580,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P590
        PCB  TYPE=TP,MODIFY=YES
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=A,KEYLEN=22
        SENSEG    NAME=PAC7TG
        PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P590,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P599
        PCB  TYPE=TP,MODIFY=YES
        PSBGEN  PSBNAME=$ROOT.P599,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P600
        PCB  TYPE=TP,MODIFY=YES
        PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=A,KEYLEN=21
        SENSEG    NAME=PAC7TD
        PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
        SENSEG    NAME=PAC7TE
        PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=A,KEYLEN=22
        SENSEG    NAME=PAC7TG
        PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P600,LANG=COBOL
        END
   :/ ADD NAME=$ROOT.P601
        PCB  TYPE=TP,MODIFY=YES
        PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
        SENSEG    NAME=PAC7TZ
        PSBGEN  PSBNAME=$ROOT.P601,LANG=COBOL
        END
   ===SEQ FOR DTM
   :/ ADD NAME=$ROOT.P610
```

```
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=A,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P610,LANG=COBOL
         END
    :/ ADD NAME=$ROOT.P620
         PCB  TYPE=TP,MODIFY=YES
         PCB     TYPE=DB,DBDNAME=PACDTD$SUF,PROCOPT=G,KEYLEN=21
         SENSEG    NAME=PAC7TD
         PCB     TYPE=DB,DBDNAME=PACDTE$SUF,PROCOPT=G,KEYLEN=17
         SENSEG    NAME=PAC7TE
         PCB     TYPE=DB,DBDNAME=PACDTG$SUF,PROCOPT=A,KEYLEN=22
         SENSEG    NAME=PAC7TG
         PCB     TYPE=DB,DBDNAME=PACDTZ$SUF,PROCOPT=A,KEYLEN=11
         SENSEG    NAME=PAC7TZ
         PSBGEN  PSBNAME=$ROOT.P620,LANG=COBOL
         END
    ===SEQ
    :/ ADD NAME=DF$ROOT.$FILE.AD
     DELETE ($INDEX..$ROOT.$FILE.AD) CL
     DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.AD)       -
                      SHR (2,3)  RUS KEY (21,0)           -
                      VOL ($VOLU)  CYL (2,1)              -
                      RECSZ (240,240) )                   -
            INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.AD)   -
                      CISZ (1024) )                       -
            DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.AD)   -
                      FSPC (10,5)                         -
                      CISZ (2048) ) /*: CATALOG ($CATU) :*/
    :/ ADD NAME=DF$ROOT.$FILE.AV
     DELETE ($INDEX..$ROOT.$FILE.AV) CL
     DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.AV)       -
                      SHR (2,3)  RUS KEY (35,4)           -
                      VOL ($VOLU)  CYL (2,1)              -
                      RECSZ (80,1100) )                   -
            INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.AV)   -
                      CISZ (1024) )                       -
            DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.AV)   -
                      FSPC (10,5)                         -
                      CISZ (2048) ) /*: CATALOG ($CATU) :*/
    :/ ADD NAME=DF$ROOT.$FILE.DD
     DELETE ($INDEX..$ROOT.$FILE.DD) CL
     DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.DD)       -
                      SHR (2,3)  RUS KEY (21,0)           -
                      VOL ($VOLU)  CYL (2,1)              -
                      RECSZ (240,240) )                   -
            INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.DD)   -
                      CISZ (1024) )                       -
            DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.DD)   -
                      FSPC (10,5)                         -
                      CISZ (2048) ) /*: CATALOG ($CATU) :*/
    :/ ADD NAME=DF$ROOT.$FILE.TD
     DELETE ($INDEX..$ROOT.$FILE.TD) CLUSTER
     DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TD)     -
                      SHR (2,3)  RUS  KEY (21,6)        -
                      VOL ($VOLU)  CYL (2,1)            -
                      RECSZ (248,248) )                 -
            INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.TD) -
                      CISZ (4096) )                     -
            DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TD) -
                      FSPC (10,5)                       -
                      CISZ (4096) ) /*: CATALOG ($CATU) :*/
    :/ ADD NAME=DF$ROOT.$FILE.TS
     DELETE ($INDEX..$ROOT.$FILE.TS) CLUSTER
     DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TS)     -
                      SHR (2,3)  RUS  KEY (21,0)        -
                      VOL ($VOLU)  CYL (2,1)            -
                      RECSZ (240,240) )                 -
            INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.TS) ) -
```

```
           DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TS) -
                     FSPC (10,5)                       -
                     CISZ (4096) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=DF$ROOT.$ROOT.TE
    DELETE ($INDEX..$ROOT.$ROOT.TE) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$ROOT.TE)     -
                     SHR (2,3)  RUS  KEYS (17,6)       -
                     VOL ($VOLV)  CYL (2,1)            -
                     RECSZ (098,098) )                 -
           INDEX   ( NAME ($INDEX..IND.$ROOT.$ROOT.TE) -
                     CISZ (4096) )                     -
           DATA    ( NAME ($INDEX..DAT.$ROOT.$ROOT.TE) -
                     FSPC (10,5)                       -
                     CISZ (4096) ) /*: CATALOG ($CATV) :*/
   :/ ADD NAME=DF$ROOT.$FILE.TG
    DELETE ($INDEX..$ROOT.$FILE.TG) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TG)     -
                     SHR (2,3)  RUS  KEYS (22,6)       -
                     VOL ($VOLU)  CYL (2,1)            -
                     RECSZ (094,094) )                 -
           INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.TG) -
                     CISZ (4096) )                     -
           DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TG) -
                     FSPC (10,5)                       -
                     CISZ (4096) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=DF$ROOT.$FILE.TU
    DELETE ($INDEX..$ROOT.$FILE.TU) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TU)     -
                     SHR (2,3)  RUS  KEYS (35 5)       -
                     INDEXED                           -
                     VOL ($VOLU)     CYL (1,1)         -
                     RECSZ (040 040) )                 -
           INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.TU) -
                     CISZ (1024) )                     -
           DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TU) -
                     CISZ (1024) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=DF$ROOT.$FILE.TV
    DELETE ($INDEX..$ROOT.$FILE.TV) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TV)     -
                     SHR (2,3)  RUS                    -
                     NONINDEXED                        -
                     VOL ($VOLU) CYL (2,1)             -
                     RECSZ (4089 4089) )               -
           DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TV) -
                     FSPC (10,5)                       -
                     CISZ (4096) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=DF$ROOT.$FILE.TW
    DELETE ($INDEX..$ROOT.$FILE.TW) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TW)     -
                     SHR (2,3)  RUS  KEY (35,4)        -
                     VOL ($VOLU)  CYL (2,1)            -
                     RECSZ (80,1100) )                 -
           INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.TW) ) -
           DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TW) -
                     FSPC (10,5)                       -
                     CISZ (4096) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=DF$ROOT.$ROOT.TZ
    DELETE ($INDEX..$ROOT.$ROOT.TZ) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$ROOT.TZ)       -
                     SHR (2,3)    KEYS (11,6)            -
                     VOL ($VOLV)  CYL (3 3)              -
                     RECSZ (8510 8510) RUS )             -
           INDEX   ( NAME ($INDEX..IND.$ROOT.$ROOT.TZ)   -
                     CISZ (4096) )                       -
           DATA    ( NAME ($INDEX..DAT.$ROOT.$ROOT.TZ)   -
                     FSPC (10,5)                         -
                     CISZ (10240) ) /*: CATALOG ($CATV) :*/
   :/ ADD NAME=DF$ROOT.$FILE.T1
    DELETE ($INDEX..$ROOT.$FILE.T1) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.T1)     -
                     SHR (2,3)  RUS  KEYS (63 5)       -
                     INDEXED                           -
                     VOL ($VOLU)     CYL (1,1)         -
```

```
                  RECSZ (068 068) )                -
         INDEX   ( NAME ($INDEX..IND.$ROOT.$FILE.T1) -
                  CISZ (1024) )                    -
         DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.T1) -
                  CISZ (1024) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=DF$ROOT.$FILE.TB
    DELETE ($INDEX..$ROOT.$FILE.TB) CL
    DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.TB)     -
                  SHR (2,3)  RUS                   -
                  NONINDEXED                       -
                  VOL ($VOLU) CYL (2,1)            -
                  RECSZ (4089 4089) )              -
         DATA    ( NAME ($INDEX..DAT.$ROOT.$FILE.TB) -
                  FSPC (10,5)                      -
                  CISZ (4096) ) /*: CATALOG ($CATU) :*/
   :/ ADD NAME=VERIFTD
    VERIFY FILE (DDTD)
   :/ ADD NAME=VERIFTS
    VERIFY FILE (DDTS)
   :/ ADD NAME=VERIFTE
    VERIFY FILE (DDTE)
   :/ ADD NAME=VERIFTG
    VERIFY FILE (DDTG)
   :/ ADD NAME=VERIFTU
    VERIFY FILE (DDTU)
   :/ ADD NAME=VERIFTV
    VERIFY FILE (DDTV)
   :/ ADD NAME=VERIFT1
    VERIFY FILE (DDT1)
   :/ ADD NAME=VERIFTB
    VERIFY FILE (DDTB)
   :/ ADD NAME=VERIFTW
    VERIFY FILE (DDTW)
   :/ ADD NAME=REPROTG
    REPRO INFILE(PAC7IG) OUTFILE(PAC7TG)
   :/ ADD NAME=REPROTB
    REPRO INFILE(INTB) OUTFILE(OUTTB)
   :/ ADD NAME=BL$ROOT.$FILE.TC
    DEFINE GENERATIONDATAGROUP -
           (NAME ($INDEXQ..$ROOT.$FILE.TC) LIMIT (3) SCR)
   :/ ADD NAME=LD$ROOT.$FILE.TG
   999999999999999999999 RECORD OF LOADING TG DATABASE
   :/ ADD NAME=LD$ROOT.$ROOT.TZ
   99999999ZZ9 RECORD OF LOADING DATABASE SPA FOR PACTABLES
   :/ ADD NAME=LD$ROOT.$FILE.TB
   99999999999999999999999999999999999999999999999999999999999999999999
   :/ ADD NAME=SY$ROOT.$FILE.TG
      GENERATE    MAXFLDS=2,MAXLITS=5
      RECORD      FIELD=(80,1,,1),
                  FIELD=(5,'     ',,81)
   :/ ADD NAME=DFSVSAM8
   4096,8
   :/ ADD NAME=DFSVSAM9
   12288,9
   :/ ADD NAME=PACTCTRL
   *      *-------------------------------------------------------*
   *      * THIS PDS IS MANAGED BY THE SYSTEMS TEAM.  IT CONTAINS *
   *      * THE DESCRIPTIONS OF MACRO-INSTRUCTIONS 'APPLCTN' AND  *
   *      * 'TRANSACT' OF THE PACTABLES SYSTEM.  THESE MUST BE    *
   *      * DEFINED IN THE I.M.S. CONTROL REGION.                 *
   *      *-------------------------------------------------------*
   *
   *-----------------------------------------------------------------------*
   * DEFINITION OF PACTABLES 2.5   DATABASES     -      CONTROL REGION - *
   *-----------------------------------------------------------------------*
      DATABASE ACCESS=UP,DBD=(PACDTD$SUF,PACDTE$SUF,PACDTG$SUF)
      DATABASE ACCESS=UP,DBD=(PACDTU$SUF,PACDTV$SUF,PACDTZ$SUF)
      DATABASE ACCESS=UP,DBD=(PACDT1$SUF,PACDTB$SUF)
   *-----------------------------------------------------------------------*
   * DEFINITION OF PACTABLES 2.5   TRANSACTIONS       - CONTROL REGION - *
   *-----------------------------------------------------------------------*
   *
   *   *-----------------------------------------------------------*
```

```
*   * WARNING: THE FOLLOWING PARAMETERS OF THE MACRO-INSTRUCTIONS  *
*   * 'TRANSACT' CANNOT BE MODIFIED:                               *
*   * 'CODE=' , 'SEGSIZE=' , 'MODE=' , 'SPA=' , 'SNGLSEG' (IN THE  *
*   * PARAMETER 'MSGTYPE=')                                        *
*   * OTHER PARAMETERS MUST BE ADJUSTED TO THE STANDARDS IN USE ON *
*   * THE INSTALLATION SITE.                                       *
*   *---------------------------------------------------------------*
   APPLCTN  PSB=$ROOT.PLNK
   TRANSACT CODE=$ROOT.CLNK,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P501
   TRANSACT CODE=$TRANT,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,     $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P601
   TRANSACT CODE=$TRANP,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,     $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P500
   TRANSACT CODE=$ROOT.C500,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P510
   TRANSACT CODE=$ROOT.C510,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P520
   TRANSACT CODE=$ROOT.C520,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P530
   TRANSACT CODE=$ROOT.C530,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P540
   TRANSACT CODE=$ROOT.C540,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P550
   TRANSACT CODE=$ROOT.C550,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P560
   TRANSACT CODE=$ROOT.C560,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P570
   TRANSACT CODE=$ROOT.C570,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P580
   TRANSACT CODE=$ROOT.C580,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P590
   TRANSACT CODE=$ROOT.C590,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P599
   TRANSACT CODE=$ROOT.C599,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P600
   TRANSACT CODE=$ROOT.C600,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
===SEQ FOR DTM
   APPLCTN  PSB=$ROOT.P610
   TRANSACT CODE=$ROOT.C610,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
   APPLCTN  PSB=$ROOT.P620
```

```
   TRANSACT CODE=$ROOT.C620,SEGSIZE=03500,MODE=SNGL,SEGNO=00050, $A
               PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,$A
               MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150,DASD)
===SEQ
*-------------------------------------------------------------------*
* DEFINITION OF BMP'S    -   PACTABLES 2.5      -  CONTROL REGION    *
*-------------------------------------------------------------------*
   APPLCTN  PSB=PTA250$SUG,PGMTYPE=BATCH
   APPLCTN  PSB=PTA300$SUG,PGMTYPE=BATCH
===SEQ
```

23.12. JCL: LOADING OF BATCH PROCEDURES

```
//$PRFJ.TPRO JOB ($CCPT),'LOADING PROCEDURES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//ET010 EXEC PGM=IEBUPDTE,PARM=NEW
//*************************************************************
//*            CATALOGING THE PACTABLES PROCEDURES            *
//*************************************************************
//*
//*    *-------------------------------------------------*
//*    *               I M P O R T A N T                 *
//*    *                  -----------                    *
//*    *      BEFORE EXECUTING THIS JOB, REPLACE ALL     *
//*    *         ':/' WITH './' USING AN EDITOR.         *
//*    *-------------------------------------------------*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$BIBP,DISP=SHR
//SYSIN    DD DATA,DLM='%%'
:/ ADD NAME=$RADP.BVTA
//*  PACTABLES 2.0   : COPY OF TD, TV DATABASES ONTO VSAM FILES
:/ ADD NAME=$RADP.PRTA
//*  PACTABLES 2.0   : PRINTING OF TABLES                        *
:/ ADD NAME=$RADP.IMTA
//*  PACTABLES 2.0   : IMPORT OF TABLES                          *
:/ ADD NAME=$RADP.EXTA
//*  PACTABLES 2.0   : EXTRACTION OF TABLES                      *
:/ ADD NAME=$RADP.GETT
//*  PACTABLES 2.0   : GENERATION OF TABLES (PACTABLES SITE)     *
:/ ADD NAME=$RADP.INTA
//*  PACTABLES 1.2   : INITIALIZATION OF PACTABLES               *
:/ ADD NAME=$RADP.LDTA
//*  PACTABLES 2.0   : TABLE DESCRIPTION LIST                    *
:/ ADD NAME=$RADP.LDTE
//*  PACTABLES 2.0   : CREATION OF THE DL/1 ERROR-MESSAGE DATABASE *
:/ ADD NAME=$RADP.LDTG
//*  PACTABLES 2.0   : LOADING OF THE DL/1 USER PARAMETER DATABASE *
:/ ADD NAME=$RADP.LDTZ
//*  PACTABLES 2.0   : LOADING OF THE DL/1 SPA DATABASE          *
:/ ADD NAME=$RADP.PMTA
//*  PACTABLES 2.0   : UPDATE OF USER PARAMETERS                 *
:/ ADD NAME=$RADP.UPTA
//*  PACTABLES 2.0   : TABLE UPDATE                              *
:/ ADD NAME=$RADP.RETA
//*  PACTABLES 1.2   : REORGANIZATION OF TABLES                  *
:/ ADD NAME=$RADP.TUTA
//*  PACTABLES 2.0   : DIRECT CONSULTATION OF TABLES             *
:/ ADD NAME=$RADP.LPTA
//*  PACTABLES 2.0   : LIST OF INSTALLED PROGRAMS                *
:/ ADD NAME=$RADP.CDT1
//*  PACTABLES 2.0   : TABLE DESCRIPTION COMPARISON              *
:/ ADD NAME=$RADP.CDT2
//*  PACTABLES 2.0   : TABLE DESCRIPTION UPDATE                  *
:/ ADD NAME=$RADP.CVTA
//*  PACTABLES 2.0   : EXTRACTION OF MODIFIED TABLE DATA         *
:/ ADD NAME=$RADP.SVTA
//*  PACTABLES 2.0   : TABLE DATABASE BACKUP (TD-TV-TG)          *
:/ ADD NAME=$RADP.RSTA
//*  PACTABLES 2.0   : RELOADING-RESTORING TABLES (TD-TV-TG DB's) *
:/ ADD NAME=$RADP.R3TA
//*  PACTABLES 2.0   : RETREVING 7.3 PACTABLES FILES             *
:/ ADD NAME=$RADP.RTTA
//*  PACTABLES 2.0   : RETRIEVAL FROM PACTABLES 7.3              *
===SEQ
%%
//
```

23.13. JCL: LOADING OF THE DBDLIB

```
//$PRFJ.TDBD  JOB ($CCPT),'DBDLIB',CLASS=$CLASSJ,
//  MSGCLASS=$MSGCL
//ET010   EXEC PGM=IEBCOPY
//*************************************************************
//*                 LOADING OF DBDLIB          PACTABLES 2.5  *
//*************************************************************
//*
//*-----------------------------------------------------------*
//*  TO BE EXECUTED ONLY IF YOUR DEVICE TYPE IS '3390' AND    *
//*  THE SUFFIX OF DBD IS '22'. OTHERWISE, YOU MUST COMPILE   *
//*  THE DBD SOURCES FOUND IN PDS PARAMETER FILE 'SY'.        *
//*-----------------------------------------------------------*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(TRK,5)
//SYSUT4   DD UNIT=$UWK,SPACE=(TRK,5)
//IND1     DD DSN=PACT.DBDLIB,DISP=OLD,UNIT=$UTAPE,LABEL=(09,SL),
//            VOL=(,RETAIN,SER=$TAPEI)
//OUT1     DD DSN=$DBDLIB,DISP=SHR
//SYSIN    DD *
    COPY INDD=((IND1,R)),OUTDD=OUT1
/*
//*
```

23.14. JCL: LOADING OF THE PSBLIB

```
//$PRFJ.TPSB JOB ($CCPT),'PSBLIB',CLASS=$CLASSJ,
//  MSGCLASS=$MSGCL
//ET010   EXEC PGM=IEBCOPY
//*************************************************************
//*                 LOADING OF PSBLIB          PACTABLES 2.5  *
//*************************************************************
//*
//*-----------------------------------------------------------*
//*  TO BE EXECUTED ONLY WHEN:                                *
//*    - YOUR CHOSEN SYSTEM ROOT IS ....: ROOT='P2'           *
//*    - THE SUFFIX OF THE DBD IS ......: SUF='22'            *
//*    - THE PRECEDING SUFFIX WAS ......: SUFR='73'           *
//*    - THE SUFFIX OF THE BATCH PSB IS : SUG='22'            *
//*  IF ONLY ONE OF THESE REQUIREMENTS IS NOT MET, YOU MUST   *
//*  COMPILE THE PDS SOURCES FOUND IN THE 'SY' PDS PARAMETER  *
//*  FILE.                                                    *
//*-----------------------------------------------------------*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(TRK,15)
//SYSUT4   DD UNIT=$UWK,SPACE=(TRK,15)
//IND1     DD DSN=PACT.PSBLIB,DISP=OLD,UNIT=$UTAPE,LABEL=(10,SL),
//            VOL=(,RETAIN,SER=$TAPEI)
//OUT1     DD DSN=$PSBLIB,DISP=SHR
//SYSIN    DD *
    COPY INDD=((IND1,R)),OUTDD=OUT1
    S M=PTABVS22
===SEQ FOR DTM
    S M=PTAD0522
    S M=PTAD1022
    S M=PTAD2022
===SEQ
    S M=PTAINI22
    S M=PTAREO22
    S M=PTARSV22
    S M=PTARSD22
    S M=PTARSG22
    S M=PTASTD22
    S M=PTASVV22
    S M=PTASVD22
    S M=PTASVG22
    S M=PTAT1022
    S M=PTAT1522
    S M=PTAT1722
    S M=PTAT1922
    S M=PTAT4022
    S M=PTAU8022
    S M=PTAXVD22
    S M=PTAXVV22
    S M=PTAXVG22
===SEQ FOR DTM
    S M=PTAV1022
    S M=PTAV2022
===SEQ
    S M=PTAXDT22
    S M=PTA10022
    S M=PTA12022
    S M=PTA15022
    S M=PTA16022
    S M=PTA25022
    S M=PTA29022
    S M=PTA30022
    S M=PTA30222
    S M=PTA31022
    S M=PTA31222
    S M=PTA32022
    S M=PTA35022
    S M=PTA36022
```

```
       S M=PTA40022
       S M=PTA42022
       S M=P2PLNK
       S M=P2P500
       S M=P2P501
       S M=P2P510
       S M=P2P512
       S M=P2P520
       S M=P2P522
       S M=P2P530
       S M=P2P540
       S M=P2P550
       S M=P2P560
       S M=P2P570
       S M=P2P580
       S M=P2P590
       S M=P2P599
       S M=P2P600
       S M=P2P601
   ===SEQ FOR DTM
       S M=P2P610
       S M=P2P620
   ===SEQ
   /*
   //*
```

23.15. JCL: LOADING OF THE LOAD-MODULES

```
//$PRFJ.TSYS JOB ($CCPT),'SYSTEM FILES LOAD',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*************************************************************
//*        LOADING OF ALL PACTABLES SYSTEM FILES              *
//*************************************************************
//*************************************************************
//*            COPY OF LOAD-MODULES IN A PDS                  *
//*************************************************************
//*
//*-----------------------------------------------------------*
//*                    I M P O R T A N T                      *
//*                    -----------------                      *
//*   IT IS NOT NECESSARY TO EXECUTE THE 'ET015' AND 'ET016'  *
//*      STEPS IF THE ON-LINE MODULES' PREFIX CONTAINS THE    *
//*      VALUE 'P2' (CHECK THE 'ROOT' 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'   *
//*                                                           *
//*   STEP 'ET017' APPLIES TO THE 'SEC' OPTION ONLY.          *
//*      (LOADING -PACSECU8-)                                 *
//*                                                           *
//*   IMPORTANT : (FOR 'SEC' VARIANT ONLY)                    *
//*   IF THE ON-LINE AND BATCH PROGRAMS ARE INSTALLED IN A    *
//*   COMMON LIBRARY, THE SELECTION OF USER'S PROGRAM         *
//*   'PACSECB' IN ONE OF THE TWO 'COPY' CLAUSES MUST BE      *
//*   CANCELLED.                                              *
//*-----------------------------------------------------------*
//ET005   EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=$OUT
//DD1      DD UNIT=$UNITP,VOL=SER=$VOLP,DISP=SHR
//SYSIN    DD *
    UNCATLG DSNAME=$MODB
    SCRATCH DSNAME=$MODB,VOL=$UNITP=$VOLP
    UNCATLG DSNAME=$MODT
    SCRATCH DSNAME=$MODT,VOL=$UNITP=$VOLP
//*
//ET007   EXEC PGM=IEFBR14
//DDA      DD DSN=$MODB,DISP=(,CATLG,DELETE),UNIT=$UNITP,
//            SPACE=(TRK,(55,10,15)),VOL=SER=$VOLP,
//            DCB=(RECFM=U,BLKSIZE=6144)
//DDB      DD DSN=$MODT,DISP=(,CATLG,DELETE),UNIT=$UNITP,
//            SPACE=(TRK,(70,10,15)),VOL=SER=$VOLP,
//            DCB=(RECFM=U,BLKSIZE=6144)
//*
//ET010  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTB     DD DSN=$MODB,DISP=SHR
//OUTP     DD DSN=$MODT,DISP=SHR
//INB      DD DSN=PACT.MBR8,DISP=OLD,UNIT=$UTAPE,LABEL=(04,SL),
//            VOL=(,RETAIN,SER=$TAPEI)
//INT      DD DSN=PACT.MTR8,DISP=OLD,UNIT=$UTAPE,LABEL=(05,SL),
//            VOL=(,RETAIN,SER=$TAPEI)
//SYSIN    DD *
    COPY INDD=((INB,R)),OUTDD=OUTB
===SEQ FOR SEC
    S M=PACSECB
===SEQ
    S M=PAP820
    S M=PAP822
    S M=PAP920
    S M=PAP922
    S M=PTABVS
===SEQ FOR DTM
```

```
        S M=PTAD05
        S M=PTAD10
        S M=PTAD20
    ===SEQ
        S M=PTAINI
        S M=PTAREO
        S M=PTARSD
        S M=PTARSV
        S M=PTARSG
        S M=PTAR20
    ===SEQ
        S M=PTASTD
        S M=PTASVV
        S M=PTASVD
        S M=PTASVG
        S M=PTATC1
        S M=PTATC2
        S M=PTATCD
        S M=PTATCV
        S M=PTATCG
        S M=PTAXVV
        S M=PTAXVD
        S M=PTAXVG
        S M=PTAT10
        S M=PTAT15
        S M=PTAT17
        S M=PTAT19
        S M=PTATIN
        S M=PTAT40
        S M=PTAT45
        S M=PTAT50
        S M=PTAU80
    ===SEQ FOR DTM
        S M=PTAV10
        S M=PTAV20
    ===SEQ
        S M=PTAXDT
        S M=PTA100
        S M=PTA120
        S M=PTA150
        S M=PTA160
        S M=PTA250
        S M=PTA290
        S M=PTA300
        S M=PTA302
        S M=PTA310
        S M=PTA312
        S M=PTA320
        S M=PTA350
        S M=PTA360
        S M=PTA400
        S M=PTA410
        S M=PTA420
        S M=PTA430
        S M=PTU001
        COPY INDD=((INT,R)),OUTDD=OUTP
    ===SEQ FOR SEC
        S M=PACSECB
    ===SEQ
        S M=PACXSY
        S M=MVSJOBB
        S M=((ZZPLNK,$ROOT.PLNK))
        S M=((ZZP500,$ROOT.P500))
        S M=((ZZP501,$ROOT.P501))
        S M=((ZZP510,$ROOT.P510))
        S M=((ZZP512,$ROOT.P512))
        S M=((ZZP520,$ROOT.P520))
        S M=((ZZP522,$ROOT.P522))
        S M=((ZZP530,$ROOT.P530))
        S M=((ZZP540,$ROOT.P540))
        S M=((ZZP550,$ROOT.P550))
        S M=((ZZP560,$ROOT.P560))
        S M=((ZZP570,$ROOT.P570))
```

```
       S M=((ZZP580,$ROOT.P580))
       S M=((ZZP590,$ROOT.P590))
       S M=((ZZP599,$ROOT.P599))
       S M=((ZZP600,$ROOT.P600))
       S M=((ZZP601,$ROOT.P601))
       S M=((ZZR980,$ROOT.R980))
   ===SEQ FOR DTM
       S M=((ZZP610,$ROOT.P610))
       S M=((ZZP620,$ROOT.P620))
   ===SEQ FOR TUF
       S M=((ZZFT00,$ROOT.FT00))
       S M=((ZZFT90,$ROOT.FT90))
   ===SEQ
   /*
   //ET015   EXEC PGM=IMASPZAP
   //SYSPRINT DD SYSOUT=$OUT
   //SYSLIB   DD DSN=$MODT,DISP=SHR
    NAME $ROOT.P501 P2P501
    VER 0005 D7F2
    REP 0005 $HEXA
    VER 01FD D7F2
    REP 01FD $HEXA
    VER 0237 D7F2
    REP 0237 $HEXA
   //*
   //ET016   EXEC PGM=IMASPZAP
   //SYSPRINT DD SYSOUT=$OUT
   //SYSLIB   DD DSN=$MODT,DISP=SHR
    NAME $ROOT.P601 P2P601
    VER 0005 D7F2
    REP 0005 $HEXA
    VER 01FD D7F2
    REP 01FD $HEXA
    VER 0237 D7F2
    REP 0237 $HEXA
   //*
   ===SEQ FOR SEC
   //ET017   EXEC PGM=IEBCOPY
   //SYSPRINT DD SYSOUT=$OUT
   //SYSUT3   DD UNIT=$UWK,SPACE=(TRK,20)
   //IND1     DD DSN=PACT.MBR8,DISP=OLD,UNIT=$UTAPE,
   //            VOL=(,RETAIN,SER=$TAPEI),LABEL=(04,SL)
   //OUT1     DD DISP=OLD,DSN=------------         <-- AUTHORIZED LIBRARY
       COPY INDD=((IND1,R)),OUTDD=OUT1
       S M=PACSECU8
   //*
   ===SEQ
```

23.16. JCL: LOADING OF DOCUMENTATION-ERROR MESSAGES

```
//$PRFJ.TTE  JOB ($CCPT),'LOADING TE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//  JCLLIB ORDER=($BIBP)
//ET010   EXEC PGM=IEHPROGM
//************************
//*
//*****  SCRATCH-UNCATLG OF THE 'TE' FILE SEQUENTIAL IMAGE
//*
//SYSPRINT DD SYSOUT=$OUT
//DD1      DD UNIT=$UNITP,VOL=SER=$VOLP,DISP=SHR
 SCRATCH DSNAME=$INDEXP..$ROOT.$ROOT.IN,VOL=$UNITP=$VOLP
 UNCATLG DSNAME=$INDEXP..$ROOT.$ROOT.IN
//*
//ET020   EXEC PGM=IDCAMS
//***********************
//*
//*****  LOADING OF THE 'TE' FILE SEQUENTIAL IMAGE
//*
//*:STEPCAT  DD DSN=$CATV,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//TEI      DD DSN=PAC.TE,DISP=OLD,UNIT=$UTAPE,LABEL=(06,SL),
//            VOL=(,RETAIN,SER=$TAPEI)
//TEO      DD DSN=$INDEXP..$ROOT.$ROOT.IN,DISP=(,CATLG,DELETE),
//            VOL=SER=$VOLP,UNIT=$UNITP,SPACE=(TRK,(20,10),RLSE),
//            DCB=(RECFM=FB,LRECL=90,BLKSIZE=6030)
//SYSIN    DD *
  REPRO INFILE (TEI) OUTFILE (TEO)
/*
//*****  LOADING OF THE 'TE' DATABASE *****
//*
//ET030   EXEC $RADP.LDTE
//*
//
```

23.17. JCL: INITIAL LOADING OF THE SPA DATABASE

```
//$PRFJ.TTZ  JOB ($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 2.5    *
//* FUNCTION.                                                     *
//*****************************************************************
//ET010   EXEC $RADP.LDTZ
//*
```

23.18. JCL: INITIAL LOADING OF THE TUF-TP DATABASE

```
//$PRFJ.TTB  JOB ($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 2.5
//* FUNCTION.
//*************************************************************
//ET010   EXEC $RADP.LDTB
//*
```

23.19. JCL: BUILDING OF THE GENERATION-FILE INDEX

```
//$PRFJ.TTC  JOB ($CCPT),'G.D.G. TC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****************************************************************
//* BUILDING GENERATION FILE INDEX 'TC'                           *
//* ----------------------------------                            *
//* JOB TO BE RUN ONLY WHEN FIRST INSTALLING THE PACTABLES 2.5    *
//* FUNCTION                                                      *
//*    . BUILDING OF DSCB MODEL                                   *
//*    . DEFINE OF GDG                                            *
//*    . FILE INITIALIZATION (USER PARAMETER BACKUP)              *
//*  ->NOTE                                                       *
//*    ----                                                       *
//*    IF "SMS" IS INSTALLED, DELETE //DD1 (ET020)                *
//*****************************************************************
//*
//ET010   EXEC PGM=IEHPROGM
//************************************************
//*       SCRATCH-UNCATLG OF DSCB                *
//************************************************
//SYSPRINT DD SYSOUT=$OUT
//DD1      DD UNIT=$UNITP,VOL=SER=$VOLP,DISP=SHR
//SYSIN    DD *
    UNCATLG DSNAME=$INDEXQ..DSCB.$ROOT.$FILE.TC
    SCRATCH DSNAME=$INDEXQ..DSCB.$ROOT.$FILE.TC,VOL=$UNITP=$VOLP
//*
//ET015   EXEC PGM=IEFBR14
//DDA      DD DSN=$INDEXQ..DSCB.$ROOT.$FILE.TC,DISP=(,CATLG,DELETE),
//            SPACE=(TRK,(0)),VOL=SER=$VOLP,UNIT=$UNITP,
//            DCB=(RECFM=VB,LRECL=1067,BLKSIZE=10674)
//************************************************
//*         BUILDING GENERATION FILES INDEX      *
//************************************************
//ET020   EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=$CATV,DISP=SHR
//*:         DD DSN=$CATU,DISP=SHR
//DD1      DD DSN=$INDEXQ..$ROOT.$FILE.TC,DISP=(,CATLG,DELETE),
//            UNIT=$UNITO,VOL=SER=$VOLO,SPACE=(TRK,0),
//            DCB=$INDEXQ..DSCB.$ROOT.$FILE.TC
//SYSIN    DD DSN=$INDEXP..$ROOT.$ROOT.SY(BL$ROOT.$FILE.TC),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//ET040   EXEC PGM=IEBGENER
//************************************************
//*               LOADING TC                     *
//************************************************
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1   DD DSN=PAC.TCENG,DISP=OLD,UNIT=$UTAPE,LABEL=(08,SL),
//            VOL=(,RETAIN,SER=$TAPEI)
//SYSUT2   DD DSN=$INDEXQ..$ROOT.$FILE.TC(+1),DISP=(,CATLG,DELETE),
//            VOL=SER=$VOLO,SPACE=(TRK,(1,1),RLSE),UNIT=$UNITO,
//            DCB=$INDEXQ..DSCB.$ROOT.$FILE.TC
//*
```

23.20. JCL: LOADING OF THE TEST FILES

```
//$PRFJ.TTAB JOB ($CCPT),'TABLES TEST FILES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//  JCLLIB ORDER=($BIBP)
//*************************************************************
//*        LOADING PACTABLES FUNCTION'S TEST FILES            *
//*************************************************************
//*
//ET010   EXEC $RADP.RSTA
//
```

23.21. JCL: COMPILATION OF ACBs

```
//$PRFJ.TACB JOB ($CCPT),'ACBGEN',CLASS=$CLASSJ,
//  MSGCLASS=$MSGCL
//ET010   EXEC ACBGEN,SOUT=$OUT,PSB='$PSBLIB',
//            DBD='$DBDLIB',ACB='$ACBLIB'
//*************************************************************
//*               COMPILATION OF ACB'S            PACTABLES  *
//*************************************************************
//*
//*-----------------------------------------------------------*
//*    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$SUF,PACDTE$SUF,PACDTG$SUF)
  BUILD DBD=(PACDTU$SUF,PACDTV$SUF,PACDTZ$SUF)
  BUILD DBD=(PACDT1$SUF,PACDTB$SUF)
  BUILD PSB=($ROOT.P500,$ROOT.P501,$ROOT.P510,$ROOT.P520)
  BUILD PSB=($ROOT.P530,$ROOT.P540,$ROOT.P550,$ROOT.P560)
  BUILD PSB=($ROOT.P570,$ROOT.P580,$ROOT.P590,$ROOT.P599)
  BUILD PSB=($ROOT.P600,$ROOT.P601,$ROOT.PLNK)
  BUILD PSB=(PTA250$SUG,PTA300$SUG)
===SEQ FOR DTM
  BUILD PSB=($ROOT.P610,$ROOT.P620)
===SEQ
/*
//*
//
```

23.22. JCL: LOADING OF THE TUF-TP MACRO-STRUCTURES

```
//$PRFJ.TCTF JOB ($CCPT),'MBR8 MTR8 PDS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*************************************************************
//*    ALLOCATION AND LOADING OF THE LIBRARY WHICH WILL CONTAINS
//*    THE MACROSTRUCTURES FOR TUF/TP (BATCH TRANSACTION FOR
//*    VISUALAGE PACBASE UPDT PROCEDURE)
//*     .STEP1 : SCRATCH UNCATLG
//*     .STEP2 : ALLOCATION
//*     .STEP3 : LOADING OF THE MACROSTRUCTURES
//*************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($MACT)
//*
//STEP2    EXEC PGM=IEFBR14
//MACT     DD DSN=$MACT,
//         DISP=(,CATLG,DELETE),UNIT=$UNITO,
//         VOL=SER=$VOLO,
//         SPACE=(6144,(20,2,2)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//*
//STEP3   EXEC PGM=IEBCOPY
//******* CHARGEMENT DES MACROS-STRUCTURES ***************
//SYSPRINT DD  SYSOUT=$OUT
//SYSUT3   DD  UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTM     DD  DSN=$MACT,                MACROS-STRUCTURES
//         DISP=OLD
//INM      DD  DSN=PACT.MACT,DISP=SHR,
//         VOL=(,RETAIN,SER=$TAPEI),UNIT=$UTAPE,LABEL=(11,SL),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//SYSIN    DD  *
  C   I=((INM,R)),O=OUTM
          S M=((FAATUFA,AATUFA))
          S M=((FAATUFL,AATUFL))
          S M=((FAATUFS,AATUFS))
          S M=((FAATUFX,AATUFX))
/*
```

23.23. LIST OF INSTALLED PROGRAMS

LIST OF INSTALLED PROGRAMS

This list is obtained through the $prfj.LPTA job which executes the $radp.LPTA procedure (see the procedure and JCL below).

It includes 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.

```
//*********************************************************************
//* PACTABLES 2.5                                                     *
//*                    LIST OF INSTALLED PROGRAMS                     *
//*********************************************************************
//$RADP.LPTA PROC FILE=$FILE,   PHYSICAL-DATABASE NUMBER
//           ROOT=$ROOT,        ROOT OF VA PAC SYSTEM
//           INDEX='$INDEX',    VSAM-FILE INDEX
//           INDEXP='$INDEXP',  INDEX OF NON-VSAM FILES
//*:         SYSTCAT='$CATV',   VSAM SYSTEM CATALOG
//           OUT='$OUT',        OUTPUT CLASS
//           OUTL='$OUT',       OUTPUT CLASS
//           MODB='$MODB',      LIBRARY OF BATCH LOAD-MODULES
//           MODT='$MODT',      LIBRARY OF ON LINE LOAD-MODULES
//           UWK=$UWK,          WORK UNIT
//           PSBLIB='$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=$IRLM
//*********************************************************************
//* INPUT :                                                           *
//* - IF A LIST OF ALL PROGRAMS IS NEEDED: ONE LINE WITH THE ROOT     *
//*   OF THE PACTABLES SYSTEM (COL.3 LENGTH 2)                        *
//* - SELECTION OF PROGRAMS IS NEEDED:                                *
//*   ONE LINE PER PROGRAM: PROGRAM CODE (COL.3 LENGTH 6)             *
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&MODB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&LPTAMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDTD     DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFTD),DISP=SHR
//*
//PACXDT  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTAXDT,PTAXDT$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&MODB,DISP=SHR
//         DD DSN=&MODT,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT  DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7TD$SUF DD DSN=&INDEX..&ROOT.&FILE.TD,DISP=SHR
//PAC7MB   DD DSN=&&LPTAMB,DISP=(OLD,DELETE)
//PAC7DS   DD SYSOUT=&OUTL
//*
```

```
//$PRFJ.LPTA JOB ($CCPT),'PROGR.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- LIST OF INSTALLED PROGRAMS  ---
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//LPTA     EXEC $RADP.LPTA
  $ROOTT
/*
```

23.24. USE TESTS

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:

. Backup (IDCAMS) TV and TD.
. Execute the reorganization (RETA) which contains:
  - Reorganization of TV (prog. PTA400 and PTA410)
  - Reorganization of TD (prog. PTA420)
  - Building of TC backup file (prog. PTA430)

. 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.

23.25. TEST JCL: INTA

```
//$PRFJ.INTA JOB ($CCPT),'INIT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*           --- TEST OF THE INTA PROCEDURE ---                      *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//INTA     EXEC $RADP.INTA
I.B.M. ESSAI                        F 1234567  ABC
```

23.26. TEST JCL: GETT

```
//$PRFJ.GETT JOB ($CCPT),'GENERATION',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- TEST OF THE GETT PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//**** INSERT HERE GETA OR GETD (SEE VA PAC 2.5)
//GETT     EXEC $RADP.GETT,MD='&&MD'
```

23.27. TEST JCL: PRTA

```
//$PRFJ.PRTA JOB ($CCPT),'PRINT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- TEST OF THE PRTA PROCEDURE ---
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//PRTA     EXEC $RADP.PRTA
 *********SUPER
EACUSTOM03101985
EATEMPER
```

23.28. TEST JCL: IMTA

```
//$PRFJ.IMTA JOB ($CCPT),'IMPORT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*          --- TEST OF THE IMTA PROCEDURE ---
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//IMTA     EXEC $RADP.IMTA,TABF='...'
 *********SUPER
 A??????
```

23.29. TEST JCL: UPTA

```
//$PRFJ.UPTA JOB ($CCPT),'UPDATE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*          --- TEST OF THE UPTA PROCEDURE ---                       *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//UPTA     EXEC $RADP.UPTA
 *********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
```

23.30. TEST JCL: SVTA

```
//$PRFJ.SVTA JOB ($CCPT),'SVTA',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- TEST OF THE SVTA PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//SVTA     EXEC $RADP.SVTA
```

23.31. TEST JCL: RSTA

```
//$PRFJ.RSTA JOB ($CCPT),'RSTA',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*           --- TEST OF THE RSTA PROCEDURE ---                      *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//RSTA     EXEC $RADP.RSTA
```

23.32. TEST JCL: RETA

```
//$PRFJ.RETA JOB ($CCPT),'REORG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- TEST OF THE RETA PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//RETA     EXEC $RADP.RETA
 *********SUPER
GACUSTOM03101985
GATEMPER
```

23.33. TEST JCL: PMTA

```
//$PRFJ.PMTA JOB ($CCPT),'PARAM.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*          --- TEST OF THE PMTA PROCEDURE ---                       *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//PMTA     EXEC $RADP.PMTA
 ********TASUPER
 ********TJ000100//$PRFJ.PRTA JOB ($CCPT),'PRTA',CLASS=$CLASSJ,
 ********TJ000200//           MSGCLASS=$MSGCL
 ********TJ000300//PRTA  EXEC $RADP.PRTA
 USER1   TAUSER1   2
```

23.34. TEST JCL: EXTA

```
//$PRFJ.EXTA JOB ($CCPT),'EXTRACTION',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- TEST OF THE EXTA PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//EXTA     EXEC $RADP.EXTA
 *********SUPER
 ACUSTOM03101985
/*
//* EXTRACTED TRANSACTIONS FILE
//PTA160.PAC7NU DD DSN=---.---.---,DISP=SHR
```

23.35. TEST JCL: TUTA

```
//$PRFJ.TUTA JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*          --- TEST OF THE TUTA PROCEDURE ---                       *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//TUTA     EXEC $RADP.TUTA
 *********SUPER
 ACUSTOM10311985
 ATEMPER
```

23.36. TEST JCL: CDT1 (DTM)

```
//$PRFJ.CDT1 JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*             --- TEST OF THE CDT1 PROCEDURE ---                    *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//CDT1     EXEC $RADP.CDT1,
//* TDMAST = 'MASTER' TABLE-DESCRIPTION FILE
//* TDSLAV = 'SLAVE' TABLE-DESCRIPTION FILE
//* XD     = EXTRACTED-DESCRIPTION FILE
// TDMAST=---.---.---,TDSLAV=---.---.---,XD=---.---.---
 *********SUPER
 ACUSTOM
 ATEMPER
/*
```

23.37. TEST JCL: CDT2 (DTM)

```
//$PRFJ.CDT2 JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*            --- TEST OF THE CDT2 PROCEDURE ---                     *
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//CDT2     EXEC $RADP.CDT2,
//* 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=---
```

23.38. TEST JCL: CVTA (DTM)

```
//$PRFJ.CVTA JOB ($CCPT),'EXPLOI.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* Pactables 2.5                                                     *
//*********************************************************************
//*********************************************************************
//*          --- TEST OF THE CVTA PROCEDURE ---
//*********************************************************************
// JCLLIB ORDER=($BIBP)
//CVTA     EXEC $RADP.CVTA,
//* TD = TABLE-DESCRIPTION FILE
//* TU = INDEX OF TV FILE
//* TV = FILE OF TABLES ASSOCIATED TO DESCRIPTIONS
//  TD=---.---.---,
//  TU=---.---.---,
//  TV=---.---.---
 *********SUPER
SACUSTOM31101985    31101987
```

23.39. PACTABLES STANDARD REINSTALLATION

STANDARD REINSTALLATION OF THE SYSTEM

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

This new version, identified by a number is made of:

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

. the product installation cardridge (or tape),

. the list of the corrected abends,

. additional instructions that might be included to complete the reinstallation steps described in this Subchapter. described in this subchapter.

Generally, only the program libraries and the error and documentation file (TE) are affected by the new version.

Therefore, re-installation most often involves the execution of the following jobs (using either the tape's JCLs or those used in previous (re-)installation):

1. COPY OF THE INSTALLATION TAPE

PACBASE0 job on the initial JCL (INST.JCL, (1,SL)).

2. RETRIEVAL OF INSTALLATION JCL

PACBASE2 job on the initial JCL (INST.JCL, (1,SL)).

NOTE: If the JCL modules required for the installation have been kept since the previous (re-)installation, the PACBASE2 job need not be executed.

It executes the MM1JCL utility with the parameters provided at installation, but to which are added the JCL module selection lines that follow:

Add in the SYSIN the following JCL module selection lines:

```
===SELM PACTSYS
===SELM PACTTE
```

ADDITIONAL selection lines may also be required: in this case, they are precised in a set of instructions provided with the tape.

The JCL modules being either obtained by this job or retrieved from previous (re-)installation, the reinstallation is carried out through the jobs described below.

3. LOADING OF BATCH AND ON-LINE LOAD-MODULES

$prfj.TSYS job (===MOD PACTSYS)

Warning: This jobs deletes all load-module libraries, allocates them and copies all programs. Two procedural methods may be used:

. execution of the complete job : In this case, programs stored in the library which do not directly come from the installation tape (user programs) must be backed up before execution of the job.

. Execution of the COPY step (IEBCOPY): In this case, it is recommended to previously delete the programs which are to be copied, in order to avoid space problems in the library.

4. RELOADING OF THE ERROR MESSAGE AND DOCUMENTATION FILE

$prfj.TTE job (===MOD PACTTE)