VisualAge Pacbase 2.5

PACTABLES 2.5 – BULL GCOS7/TDS OPERATIONS MANUAL

DETD7000251A

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.

First Edition (April 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 the Pactables Release 2.5 function.

HOW TO USE THIS MANUAL FOR SYSTEM INSTALLATION

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

. The 2.5 Release is 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 of existing data from the former release.

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. PACTABLES PARAMETERS

PACTABLES PARAMETERS

The JCL provided at installation time includes parameters (a list is included below) to set up coding standards and to dispatch files among the site's disks.

Parameters are coded '$XXXXX'. The '$' character points out the parameter in the processed element's name and 'XXXXX' is its code.

Parameters are substitued globally on the JCL when 'PTZZEXEC' and 'PTZZJCL' procedures are run. These procedures are detailed in subchapter 'JCL INSTALLATION', in chapter 'INSTALLATION'.

In the continuation of this manual, only the parameterized names of libraries and files will be used.

```
COMM 'VA-PACTABLE 2.5     ';
*****************************************
*                                       *
*        INSTALLATION PARAMETERS        *
*                                       *
*  REPLACE, IF NEEDED, THE DEFAULT      *
*  VALUE OF EACH PARAMETER.             *
*  EACH PARAMETER LINE IS FORMATTED AS  *
*  FOLLOWS:                             *
*    $NNNNN = VALUE                     *
*                                       *
*                                       *
*  - THE EQUAL SIGN (DELIMITER) AND     *
*    THE AMPERSAND CANNOT BE USED IN    *
*    A PARAMETER VALUE.                 *
*                                       *
*  SUBSEQUENTLY,                        *
*                                       *
*  - ALL LINES WHOSE FIRST NON-BLANK    *
*    CHARACTER IS NOT A DOLLAR SIGN     *
*    ARE CONSIDERED AS COMMENTS.        *
*                                       *
*  - THE EQUAL SIGN (DELIMITER)         *
*    CANNOT BE USED IN A PARAMETER      *
*    VALUE.                             *
*                                       *
*****************************************

******** LANGAGE CODE                   *
$LANG   =  E
******** DEFAULT PACTABLE USER NAME     *
$USER   =  CGI
******** INSTALLATION CATALOG NAME      *
$CTNM   =  PT
******** TDS PACTABLE NAME              *
$NMTD   =  TDST
******** NAME OF INSTALLATION TAPE      *
$TAPE   =  XXXXXX
******** DEVICE OF INSTALLATION TAPE      *
$DVTP   =  CT/M5
************ SUFFIX OF LIBRARIES **********
******** CU BATCH
$LIBCUBT  =  CUBLIBT
******** CU TP
$LIBCUTT  =  CUTLIBT
******** PERMANENT CU
$LIBCUPT  =  CUPLIBT
******** CU OF MODULES CALLING SUB-PROGRAMS GENERATE WITH
******** SIECLE OPTION - MUST BE DIFFERENT OF $LIBCUBT & $LIBCUTT
$LIBCU20  =  CU20LIB
******** JCL
$LIBJCLT  =  JCLLIBT
******** CALL OF JCL PROCEDURE
$LIBINVT  =  INVLIBT
******** PRINT OF JCL IN REPORTS
**       $LIST = SPACE  NO PRINT
**       $LIST = &LIST     PRINT
$LIST     =  &LIST
******* LM
$LIBLMT   =  LMLIBT
******* SM     (SEE NOTE)
$LIBSMT   =  SMLIB
******* USER SL
$LIBSUT   =  SULIBT
******* TDS SL (SEE NOTE)
$LIBSLT   =  SLLIB
******** RADICAL-IDENTIFIER OF PACTABLE FILES **
$TRTAB  =  PT200
******** PREFIX OF PACBASE BATCH USER FILES
** THE VALUE OF $TRPAC MUST BE EQUAL TO THAT OF $NMBU OF PACBASE
$TRPAC  =  PT.PB200.BU
*******  CISIZE OF BATCH FILES
*******  DEFAULT VALUE FOR DISC MS/D500
```

```
$CISEQ  =  14336
*******  MEDIA TYPE OF SEQUENTIAL FILE TC  *****
*** IF CATALOGED FILE
*** VALUES OF $MDSV : T FOR TAPE OR D FOR DISK
$MDSV   =  D
*************** REFERENCES **************
*NMXX   =  PREFIX FILE
*DVXX   =  DEVICE FILE
*MDXX   =  MEDIA  FILE
*CTXX   =  CATALOG (Y OR N)
******** TDS  FILES
$DVTD   =  MS/D500
$MDTD   =  DISC01
$CTTD   =  Y
******** TDS USER FILES
$NMTU   =  PTTU
$DVTU   =  MS/D500
$MDTU   =  DISC02
$CTTU   =  Y
******** BATCH USER FILES
$NMBU   =  PTBU
$DVBU   =  MS/D500
$MDBU   =  DISC03
$CTBU   =  Y
******** TEMPORARY FILES
$DVTM   =  MS/D500
$MDTM   =  DISC04
******** LIBRARIES
$NMLI   =  PTLI
$DVLI   =  MS/D500
$MDLI   =  DISC05
$CTLI   =  Y
************* LINK ENVIRONMENT (SEE NOTE) *
******** NAME OF PACTABLE LINK TPR0       *
$TPR0   =  TPR
*****************************************
```

PARAMETER FILES

The following tables present all files, grouped by family (first parameter of their external name). The user can then choose the value of the parameters and assess their repercussion on the names of the Pactables system files.

LIBRAIRIES

before parameterizing	with default values
$NMLI.$LIBCUBT	PTLI.CUBLIBT
$NMLI.$LIBCUPT	PTLI.CUPLIBT
$NMLI.$LIBCUTT	PTLI.CUTLIBT
$NMLI.$LIBCU20	PTLI.CUT20
$NMLI.$LIBINVT	PTLI.INVLIBT
$NMLI.$LIBJCLT	PTLI.JCLLIBT
$NMLI.$LIBLMT	PTLI.LMLIBT
$NMLI.$LIBINVT	PTLI.INVLIBT
$NMLI.$LIBSUT	PTLI.SULIBT
$NMTD.$LIBSM	TDST.SMLIB
$NMTD.$LIBSL	TDST.SLLIB

USER BATCH FILES

before parameterizing	with default values
$NMBU.$TRTABTC	PTBU.PT200TC

USER TDS FILES

before parameterizing	with default values
$NMTU.$TRTABTD	PTTU.PT200TD
$NMTU.$TRTABTG	PTTU.PT200TG
$NMTU.$TRTABTV	PTTU.PT200TV
$NMTU.$TRTABTE	PTTU.PT200TE

2.3. THE ON-LINE PROGRAM LIBRARY

THE SHARABLE MODULES LIBRARY: $NMTD.$LIBSMT

Its size is about 300 blocks of 1024 characters,
it includes the following programs:

CODE	OPERATION AND MEANING
XXPAA0	First and last TPR
XXPLNK	Interface between user pgm and Pactables
XXP500	Access, FT or 'clear': initial screen
XXP510	C1 : mono-item consultation and CR, CM, MO, AN for update
XXP520	C2 : multi-item consultation and AN for deletion
XXP530	LT : tables list
XXP540	LS : sub-schemas/sub-systems list
XXP550	LD : documentation list
XXP560	C3 : consultation of an archived item
XXP570	help screen
XXP580	LH : list of historical accounts
XXP590	LJ, LE : print request
XXP599	Display of system errors
XXP600	Update of passwords and language
XXP610	User codes update
XXP620	Access authorizations update

The XX prefix takes the following values:

- ZT for native Pactables.

IMPORTANT NOTE

Two additional programs (P510 and P520) are supplied with release 2.0 in the $NMLI.$LIBCU20 library.

In update operations, the P510 and P520 programs may call user check sub-programs to perform additional checks. The default generation option of these sub-programs is 'without century-management'.

From release 2.0 and higher, if user check sub-programs are generated with the century management option, the new P510 and P520 programs must be linked instead of the ones supplied at installation (in $TPR0 of $NMTD.$LIBSMT). In all cases, ALL user check sub-programs must be generated with the same option.

2.4. THE BATCH PROGRAM LIBRARY

THE LOAD-MODULE LIBRARY: $NMLI.$LIBLMT

Its size is about 800 blocks of 1024 characters,
it includes the following programs:

CODE	PROCEDURES	MEANING
PTAINI	INTA	File initialization
PTAU80	TUTA	Optimized production turnover
PTA100	PMTA	Parameter update
PTA120	--	-- -- --
PTA150	EXTA	Table extraction
PTA160	--	-- -- --
PTA250	GETT	Table generation
PTA290	-- LDTA	Table / List generation
PTA300	UPTA	table update
PTA310	IMTA	Table Import
PTA320	PRTA	Table printing
PTA350	-- UPTA	-- --
PTA360	-- --	-- --
PTA400	RETA	Table reorganization
PTA410	--	-- -- --
PTA420	--	-- -- --
PTA430	--	-- -- --
PTAD05	CDT1	Table-content comparison
PTAD10	--	-- --
PTAD20	CDT2	-- --
PTAR20	R2TA	Table retrieval 802, 1.2 --> 2.5
-	R3TA	-- --
PTARTG	R2TA	Retrieval 7.2 ---> 8.0
PTARSD	RSTA	Restoration of TD
PTARSG	--	-- of TG
PTARSV	--	-- of TV

CODE	PROCEDURES	MEANING
PTASVD	SVTA	Backup of TD
PTASVG	--	-- TG
PTASVV	--	-- TV
PTAV10	CVTA	Table comparison
PTAV20	--	-- --
PTAXVD	R2TA R3TA	Retrieval 7.2 or 7.3 ---> 2.5
PTAXVG	--	-- TG
PTAXVV	--	-- TV

IMPORTANT NOTE

An additional version of programs PTA300 and PTA310 is supplied with release 2.0.

In update operations, the PTA300 and PTA310 programs can call user check sub-programs to perform additional checks. The default generation option of these sub-programs does not include the century.

From release 2.0 and higher, if user check sub-programs are generated with the century management option, the two new programs (PTA300 and PTA310) supplied in the '$NMLI.LIBCU20' library must be linked instead of the programs supplied at installation in the UPTA and IMTA procedures.
In all cases ALL user check sub-programs must be generated with the same option.

2.5. THE OTHER LIBRARIES

THE SOURCE LIBRARY: $NMLI.$LIBSLT

Its size is about 1 cylinder. It includes the TDS source and source of the Pactables conversation and end TPR. It also includes the macrostructures required to operate the TUF facility.

The Macro-structures are the following ones:

CODE	MEANING	SL MEMBER
AATUFA	Description of table Data Element	TUFUPDTA
AATUFL	'LT' or 'LH' list	TUFUPDTL
AATUFS	'LS' or 'LC' list	TUFUPDTS
AATUFX	List of items	TUFUPDTX

These macro-structures are used in user on-line application programs using the TUF-TP facility.
They are used to add the description of communication areas which are necessary to the call of TUF900 sub-program in the TUF-TP facility.

These Macro-structures are supplied as VA Pac updating 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.

THE ON-LINE COMPILE-UNIT LIBRARY: $NMLI.$LIBCUTT

It includes all on-line program compile-units as well as the following sub-programs:

CODE	MEANING
ZAR980	Message formatting (*)
ZAR985	Recognition of terminal type (*)
PAP830	Optimized general access module
PAP930	User general access module
ZTACCE	Table file access sub-program (TP)
TUF000	TUF sub-program called by user program
TUF900	TUF Table access sub-program (TP)

THE BATCH COMPILE-UNIT LIBRARY: $NMLI.$LIBCUBT

It includes all batch program compile-units as well as the following sub-programs:

CODE	MEANING
PACABE	Abnormal execution report
PACSEP	Separator page printing
PTA800	Optimized access module
PTA900	Generalized access module
PTACCE	(Batch) table file access sub-program

THE PERMANENT COMPILE-UNIT LIBRARY: $NMLI.$LIBCUPT

Its size is about 1 cylinder. It includes:

- the ZAR980, ZAR985, PAP830, PAP930, ZTACCE sub-programs which are to be linked to the TDS.

- the PTA800 and PTA900 generalized access sub-programs which are to be used in batch user applications accessing Pactables.

2.6. SYSTEM FILES

THE 'SYSTEM' FILES

They represent the system itself. They are not modified by daily handling and they must be re-loaded if the system has to be re-installed. These are:

. The 10 librairies previously defined:

. $NMLI.$LIBCUBT

. $NMLI.$LIBCUPT

. $NMLI.$LIBCUTT

. $NMLI.$LIBCU20

. $NMLI.$LIBINVT

. $NMLI.$LIBJCLT

. $NMLI.$LIBLMT

. $NMLI.$LIBSUT

. $NMTD.$LIBSMT

. $NMTD.$LIBSLT

. A Pactables error message and documentation file:

. External name: $NMTU.$TRTABTE
. Size : about 900 records
. Organization : UIND
. Recsize : 90
. Key : 17 (position 1)
. Use : Batch and on-line

2.7. EVOLVING FILES

EVOLVING FILES

They contain the user's data. They are processed by the system in either on-line or batch mode.

the first two parameters make up the actual Pactables files.

. Table description file (TD)

```
. External name: $NMTU.$TRTABTD
. Organization : UIND
. Recsize      : 240
. CI size      : 4096
. Key          : 21 (position 1)
. Use          : Batch and on-line
. Space        : 16 record per C.I. of 4096
```

. Table contents file (TV)

```
. External name: $NMTU.$TRTABTV
. Organization : UIND
. Recsize      : 80 to 1059
. CI size      : 4096
. Key          : 35 (position 5)
. Use          : Batch and on-line
. Space        : varies with the table size.
```

The third file contains the user parameters required for the system operations. It is managed through a specific batch procedure.

. user parameter file (TG)

```
. External name: $NMTU.$TRTABTG
. Organization : UIND
. Recsize      : 85
. CI size      : 4096
. Key          : 22 (position 1)
. Use          : Batch and on-line
```

The TG file includes:

. User codes and corresponding access authorizations,
. Table printing user JCL.

Pactables files can be saved in the following file:

. Generation backup:

```
. External name: $NMBU.$TRTABTC
. Organization : USEQ
. Recsize      : 1063 maxi
. Use          : Batch
```

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 necessary to run the system.

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

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 in use is TDS/GCOS 7.

The table contents file (TV), user parameter file (TG) and table description file (TD) are updated in on-line mode. Therefore, they must be protected by the TDS archiving option (Journal Before).

The average size of on-line programs is approximately 50Kb.

GENERAL INFORMATION - HOW THE SYSTEM RUNS

General characteristics of the system:

. There are two transaction codes.

PT00: to use the tables,

PT90: to update user passwords and language code.

Both codes point to TPR 'xxPAA0'.

Both transaction code values are set by the user according the following considerations: the fourth character of the transaction code determines the type of terminal in use:

```
1:   only VIP7700 screens may be used
2:   only IBM3270 screens may be used
3:   only VIP7800 screens may be used

Any other value: QUESTAR screens may be used

EXAMPLE:

     PT00, PT90  ----> QUESTAR
     PT01, PT91  ----> VIP7700
     PT02, PT92  ----> IBM3270
     PT03, PT93  ----> VIP7800
```

A message-formatting program (ZAR980) is integrated in the Pactables function. It must be linked, when the TDS is being generated, via the introduction of the command 'USE ZAR980' in the TDS source.

This command is useless with Pactables-FORMS since FORMS manages screen types.

Each conversation starts and ends with the execution of TPR 'xxPAA0'.

Its source code is supplied to enable the user to insert in standard any conversation beginning and end processing on the site or to modify the standard transaction codes for convenience purposes. These codes are:

```
PT00, PT90 ---> QUESTAR mode
PT01, PT91 ---> VIP7700 mode
```

. When an abend is managed by the VA Pac System, a MAP ABEND is displayed. The display program of this screen (xxP599) is called by an 'ABORT' call.

. 'FT' entered in the OPERATION field on the Pactables initial screen ensures a correct exit.

3.3. ACCESS METHODS

ACCESS METHODS

Files are managed with the UFAS indexed access method without secondary index.

All batch procedures include a DEALLOC/PREALLOC step in case files are re-loaded.

Files are protected against concurent read-write accesses.

3.4. BATCH ENVIRONMENT

BATCH ENVIRONMENT

In batch mode, the system uses standard functions of the operating system and the UFAS access method.

The size of the memory necessary for the execution of the batch procedures varies according to the size of the buffers allocated to the files that these procedures use.

FILE SIZE

In order to estimate the amount of disk space required by the Pactables System, the following charts list each catalog and file with its size (these are the installation default values).

The maximum global size of the Pactables System is 4 million bytes, taking into account installation default values.

THE SYSTEM FILES

Parameterized names	Contents	Size (bytes)
$NMLI.$LIBLMT	Batch programs	800,000
$NMTD.$LIBSM	On-line programs	300,000
$NMLI.$LIBSU	Backup library	100,000
$NMTD.$LIBSL	Sources library	100,000
$NMLI.$LIBJCLT	JCLs library	150,000
$NMTU.$TRTABTE	Error messages	100,000
	TOTAL	: 1,550,000

EVOLVING FILES

Parameterized names	Contents	Size (bytes)
databases		
$NMTU.$TRTABTD	Descriptions (for 1000 descriptions)	250,000
$NMTU.$TRTABTV	Contents (for 2000 tables 120 ch.)	250,000
$NMTU.$TRTABTG	User parameters (for 1000 parameters)	100,000
$NMTU.$TRTABAD	Optimized descriptions	250,000
$NMTU.$TRTABAV	Optimized contents	250,000
	TOTAL	: 1,100,000

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.

. A description of steps:

- symbolics or parameters used,

- list of the files used (temporary or permanent),

- possible 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)

.Table generation (GETT)

.Table Update (UPTA)

.Table printing (PRTA)

.Table importation (IMTA)

.Table reorganization (RETA)

.Table backup (SVTA)

.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, 8.xx or 1.2 Pactables files(R3TA)

NOTE

Pactables does not provide a journal of update transactions.

EXECUTION JCL

Procedure names are made up of two parts.

- The first four characters represent the procedure type:

. PTEX operations procedures

. PTIN installation procedures

. PTUS user procedures

. PTZZ JCL interpretation

- The last four characters indicate the procedure.

```
    examples: . PMTA user parameter update
              . EXTA table extraction
```

All JCLs include parameters. Some parameters, which are common to all procedures, do not appear in the symbolics in use for each procedure. These are the names of the physical media.

All batch procedures are initiated by INVOKE from a PTIVXXXX member ($NMLI.$LIBINVT library); to each procedure corresponds such a member.

example : . PTIVPRTA call of the printing procedure

These members include a call to the JCL to be submitted (INVOKE), and for the batch procedures with user input, the contents of this input on 80-character-long data lines bounded by $INPUT and $ENDINPUT.

4.3. ABNORMAL EXECUTIONS

ABNORMAL EXECUTION

A batch program execution may terminate abnormally.

For example, input-output errors on the system files or on the database trigger the printing of a report (PAC7EI) which lists the errors encountered and the setting of SWITCH-20.

In most cases, this SWITCH-20 and the type of operation will help determine the cause of the abnormal end, for example, resources not available, file too small, etc.

If there is no PAC7EI report and if the ABORT code indicates a problem with VA Pac programs, it is necessary to contact IBM technical support. The corresponding listings should be saved for problem-solving purposes.

5. TABLE INITIALIZATION (INTA)

5.1. INTRODUCTION

INTA: TABLE INITIALIZATION

INTRODUCTION

This procedure initializes the table description 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).

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: CREATE

INITIALIZATION OF FILES: PTAINI

```
.Input file

.Output files:
 -Table Descriptions File
  EFN : $NMTU.$TRTABTD
 -Table Contents File
  EFN : $NMTU.$TRTABTV

.Output report:
 -Initialization review
```

5.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*              TABLE FILES INITIALIZATION           *';
COMM '*              ==========================           *';
COMM '*                                                   *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
CR   IF=*INTA,
     OF=(TMBINTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : TD TV ***';
IV   PTINALTD ($NMLI.$LIBJCLT,&RFLI);
IV   PTINALTV ($NMLI.$LIBJCLT,&RFLI);
COMM '*** PTAINI ***';
STEP PTAINI,FILE=($NMLI.$LIBLMT,&RFLI),REPEAT,DUMP=DATA;
     SZ  50;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU;
     ASG PAC7MD,TMBINTA,TEMPRY,&RFTM;
     ASG PAC7ED,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXINTA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

6. TABLE GENERATION (GETT)

6.1. INTRODUCTION

GETT: TABLE GENERATION

INTRODUCTION

This procedure updates the Table-Descriptions file using the table descriptions extracted from the VisualAge Pacbase Database, and initializes the generated tables in the Table-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 table descriptions used in input by the GETT procedure.

The TD and TV files being updated by this procedure, access to on-line use must therefore be closed except if the material in use allows Batch/TP concurrency.

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.

6.2. DESCRIPTION OF STEPS

GETT: DESCRIPTION OF STEPS

UPDATE OF TABLE FILES: PTA250

```
.Permanent input-output files:
 -Table-description file
  EFN : $NMTU.$TRTABTD
 -Table-contents file
  EFN : $NMTU.$TRTABTV

.Input transaction file (GETD or GETA output):
 -Update transactions
  EFN : $TRPAC.GETA&USER

.Output file
  EFN : TPAC7TK

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

PRINTING OF DESCRIPTIONS: PTA290

```
.Permanent input file:
 -Table-description file
  EFN : $NMTU.$TRTABTD

.Transaction input file:
 -Print request
  EFN : TPAC7TK

.Output report:
 -Printout of descriptions
```

6.3. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*                TABLE GENERATION                   *';
COMM '*                ================                   *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '* &USER   : USER CODE FOR USER INPUT FILE SUFFIX     *';
COMM '*           ($INDBU.GETA.&USER)                     *';
COMM '* &SIZEWK : WORK FILE SIZE IN TRACKS          (15)  *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    USER='$USER',SIZEWK=15;
OVL HOLD;
COMM '*** PTA250 ***';
STEP PTA250,FILE=($NMLI.$LIBLMT,&RFLI),REPEAT,DUMP=DATA;
     SZ  50;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU;
     ASG PAC7MD,$TRPAC.GETA&USER,&RFBU;
     ASG PAC7TK,TPAC7TK,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TK,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TK,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
     ASG PAC7ET,SYS.OUT;
     SWK WKDISK=(SZ=05,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA290 ***';
STEP PTA290,FILE=($NMLI.$LIBLMT,&RFLI),REPEAT,DUMP=DATA;
     SZ  50;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TPAC7TK,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MB,NBBUF=1;
     ASG PAC7ID,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXGETT - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

7. TABLE UPDATE (UPTA)

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

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 to run 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, PTA300, has been shipped with Pactables Rel. 2.0. (in $NMLI.$LIBCU20).

During updates, the PTA300 program may call the user check routines in order to perform additional checks. The default generation option for these sub-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, PTA300, must therefore be linked (in $NMLI.*LIBLMT) and used instead of the PTA300 program of the installation.

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

7.2. 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 '/' be 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

7.3. DESCRIPTION OF STEPS

UPTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

TABLE UPDATE: PTA300

```
.Permanent input files:
 -Table-description file
  PAC7TD : $NMTU.TRTABTD
 -Error-message file
  EFN : $NMTU.TRTABTE
 -User parameters file
  EFN : $NMTU.TRTABTG

.Permanent input-output file:
 -Table contents file
  EFN : $NMTU.TRTABTV

.Input transaction file:
 -Update transactions
  EFN : TMBUPTA

.Output file:
 -Print requests
  EFN : TPAC7DE

.Output report:
 -Transaction review

.Work file:
 -Prepared transactions
  EFN : TPAC7MT
```

FORMATTING FOR PRINTING: PTA350

.Permanent input files:
 -Table-description file
  EFN : $NMTU.TRTABTD
 -Table-contents file
  EFN : $NMTU.TRTABTV

.Input transaction file:
 -Print request
  EFN : TPAC7DE

.Output file:
 -Print file
  EFN : TPAC7ET

.Output report:
 -Statistics on printing

PRINTING OF TABLES: PTA360

.Permanent inpput file:
 -Table-description file
  EFN : $NMTU.TRTABTD

.Input Transaction file:
 -Print file
  EFN : TPAC7ET

.Output report:
 -Printing of tables

7.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*            UPDATE                                 *';
COMM '*            ======                                 *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '* &SIZEWK : WORK FILE SIZE IN TRACKS           (10)  *';
COMM '* &SIZEED : REPORT FILE SIZE IN TRACKS         (10)  *';
COMM '*                                                   *';
COMM '*  &1     : USER INPUT ORIGIN                        *';
COMM '*  - BY DEFAULT &1=JCL                              *';
COMM '*       INVOKE PROCEDURE PARAMETERS                 *';
COMM '*  - OTHERWISE  &1 = EXTA, CVTA                     *';
COMM '*       PTUSEXTA OR PTEXCVTA OUTPUT RESULT STORED IN *';
COMM '*       $NMLI.$LIBSUT LIBRARY                       *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL JCL,USER='$USER',SIZEWK=10,SIZEED=10,
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
JUMP CR&1;
CRJCL:
CR   IF=*UPTA,
     OF=(TMBUPTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
JUMP CREND;
CREXTA:CRCVTA:
CR   IF=($NMLI.$LIBSUT,&RFLI,SUBFILE=MBUPTA_&1&USER),
     OF=(TMBUPTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80),
     COMFILE=($NMLI.$LIBJCLT,&RFLI,SUBFILE=PTEXPDSL),START=2;
CREND:
COMM '*** PTA300 ***';
STEP PTA300,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7MS,TMBUPTA,TEMPRY,&RFTM;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM,END=PASS;
     ALC PAC7DE,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7DE,NBBUF=1;
     ASG PAC7MT,TPAC7MT,TEMPRY,&RFTM;
     ALC PAC7MT,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7MT,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA350 ***';
STEP PTA350,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
```

```
        ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM;
        DEF PAC7DE,NBBUF=1;
        ASG PAC7ET,TPAC7ET,TEMPRY,&RFTM,END=PASS;
        ALC PAC7ET,SZ=&SIZEED,UNIT=TRACK,INCRSZ=01;
        DEF PAC7ET,NBBUF=1;
        ASG PAC7EX,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
    ESTP;
    JUMP ERR,SW20,EQ,1;
    COMM '*** PTA360 ***';
    STEP PTA360,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
        SZ  60;
        ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
            SHARE=MONITOR;
        DEF PAC7TD,NBBUF=1,READLOCK=STAT;
        ASG PAC7ET,TPAC7ET,TEMPRY,&RFTM;
        DEF PAC7ET,NBBUF=1;
        ASG PAC7EY,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        SWK WKDISK=(SZ=05,&RFTM);
    ESTP;
    JUMP ERR,SW20,EQ,1;
    JUMP END;
    ERR:
    SEND ' PTUSUPTA - ABNORMAL END OF RUN (I/O ERROR) ';
    LET  SEV 3;
    END:
```

8. TABLE PRINTING (PRTA)

8.1. INTRODUCTION

PRTA: TABLE PRINTING

INTRODUCTION

This procedure performs a batch print 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.

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 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

8.3. DESCRIPTION OF STEPS

PRTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

EXTRACTION OF TABLES FOR PRINTING: PTA320

```
.Permanent input files:
 -Table-description File
  EFN : $NMTU.TRTABTD
 -Error-message file
  EFN : $NMTU.TRTABTE
 -Table-contents File
  EFN : $NMTU.TRTABTV
 -User-parameter file
  EFN : $NMTU.TRTABTG

.Input transaction file:
 -Update transactions
  EFN : TMBPRTA

.Output file:
 -Print requests
  EFN : TPAC7DE

.Output report:
 -Transaction review
```

PREPARATION FOR PRINTING: PTA350

.Permanent input files:
 -Table-description File
  EFN : $NMTU.TRTABTD
 -Table-contents file
  EFN : $NMTU.TRTABTV

.Input transaction file:
 -Print requests
  EFN : TPAC7DE

.Output file:
 -Print file
  EFN : TPAC7ET

.Output report:
 -Statistics on printing

PRINTING OF TABLES: PTA360

.Permanent input file:
 -Tables-description file
  EFN : $NMTU.$TRTABTD

.Input transaction file:
 -Print file
  EFN : TPAC7ET

.Output report:
 -Printing of tables

8.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*            TABLE PRINTING                         *';
COMM '*            ==============                         *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '* &SIZEED : WORK FILE SIZE IN TRACKS          (10)  *';
COMM '* &SIZEET : REPORT FILE SIZE IN TRACKS        (10)  *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    SIZEED=10,SIZEET=10;
OVL HOLD;
CR  IF=*PRTA,
    OF=(TMBPRTA,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTA320 ***';
STEP PTA320,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7CA,TMBPRTA,TEMPRY,&RFTM;
     DEF PAC7CA,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
     ASG PAC7XE,SYS.OUT;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM,END=PASS;
     ALC PAC7DE,SZ=&SIZEED,UNIT=TRACK,INCRSZ=01;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA350 ***';
STEP PTA350,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM;
     DEF PAC7DE,NBBUF=1;
     ASG PAC7ET,TPAC7ET,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ET,SZ=&SIZEET,UNIT=TRACK,INCRSZ=01;
     DEF PAC7ET,NBBUF=1;
     ASG PAC7EX,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA360 ***';
STEP PTA360,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7ET,TPAC7ET,TEMPRY,&RFTM;
     DEF PAC7ET,NBBUF=1;
```

```
        ASG PAC7EY,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        SWK WKDISK=(SZ=05,&RFTM);
    ESTP;
    JUMP ERR,SW20,EQ,1;
    JUMP END;
    ERR:
    SEND ' PTUSPRTA - ABNORMAL END OF RUN (I/O ERROR) ';
    LET  SEV 3;
    END:
```

9. TABLE IMPORT (IMTA)

9.1. 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 performed 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 if the material in use allows batch/TP concurrency.

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

If the table to be imported is large, you may have to --prior to this procedure execution-- save and reload, to either increase the size of TV file, or physically reorganize this file so as to make all the 'free space' initially provided available.

RESTRICTION

Each execution of the procedure allows you to import only one table.

IMPORTANT NOTE:

An alternative version of the update program, PTA312, is shipped with Pactables Rel. 2.0 (in $NMLI.$LIBCU20).

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

From Release 2.0 and higher, if the user check routines are generated with the century-management option, the new program, PTA31O, must be linked (in $NMLI.$LIBLMT) and used instead of the PTA310 program created at installation.

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

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 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)

9.3. DESCRIPTION OF STEPS

IMTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

TABLE CHECK AND UPDATE: PTA310

```
.Permanent input files:
 -Table-description file
  EFN : $NMTU.$TRTABTD
 -Error-message file
  EFN : $NMTU.$TRTABTE
 -User-parameter file
  EFN : $NMTU.$TRTABTG

.Permanent input-output file:
 -Table-contents files
  EFN : $NMTU.$TRTABTV

.Input file:
 -Request transactions
  EFN : TMBIMTA
 -External table-file
  EFN : TMBTABF

.Output file:
 -Print requests
  EFN : TPAC7DE

.Output report:
 -Execution report
```

FORMATTING OF PRINTOUT: PTA350

.Permanent input files:
 -Table-descriptions file
  EFN : $NMTU.$TRTABTD
 -Table-contents file
  EFN : $NMTU.$TRTABTV

.Input transaction file:
 -Print requests
  EFN : TPAC7DE

.Output file:
 -Print file
  EFN : TPAC7ET

.Output report:
 -Printing statistics

PRINTING: PTA360

.Permanent input file:
 -Table-description file
  EFN : $NMTU.$TRTABTD

.Input transaction file:
 -Print file
  EFN : TPAC7ET

.Output report:
 -Table printout

9.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*            TABLE IMPORT                           *';
COMM '*            ============                           *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '* &TABF   : TABLE FILE NAME TO BE IMPORTED          *';
COMM '* &SIZEED : WORK FILE SIZE IN TRACKS          (10)  *';
COMM '* &SIZEET : REPORT FILE SIZE IN TRACKS        (10)  *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL SIZEED=10,SIZEET=10,TABF=$NMBU.IMTA,
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
CR  IF=*IMTA,
    OF=(TMBIMTA,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
CR  IF=(&TABF,&RFBU),
    OF=(TMBTABF,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=$CISEQ,RECSZ=999,RECFORM=FB);
COMM '*** PTA310 ***';
STEP PTA310,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7MV,TMBIMTA,TEMPRY,&RFTM;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7NK,TMBTABF,TEMPRY,&RFTM;
     DEF PAC7NK,NBBUF=1;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM,END=PASS;
     ALC PAC7DE,SZ=&SIZEED,UNIT=TRACK,INCRSZ=01;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA350 ***';
STEP PTA350,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM;
     DEF PAC7DE,NBBUF=1;
     ASG PAC7ET,TPAC7ET,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ET,SZ=&SIZEET,UNIT=TRACK,INCRSZ=01;
     DEF PAC7ET,NBBUF=1;
     ASG PAC7EX,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA360 ***';
STEP PTA360,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
```

```
        ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
            SHARE=MONITOR;
        DEF PAC7TD,NBBUF=1,READLOCK=STAT;
        ASG PAC7ET,TPAC7ET,TEMPRY,&RFTM;
        DEF PAC7ET,NBBUF=1;
        ASG PAC7EY,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        SWK WKDISK=(SZ=05,&RFTM);
    ESTP;
    JUMP ERR,SW20,EQ,1;
    JUMP END;
    ERR:
    SEND ' PTUSIMTA - ABNORMAL END OF RUN (I/O ERROR) ';
    LET  SEV 3;
    END:
```

10. TABLE REORGANIZATION (RETA)

10.1. INTRODUCTION

RETA: TABLE REORGANIZATION

INTRODUCTION

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

RETA deletes the records that were logically deleted during update and reorganizes these files' historical accounts according to user 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.

10.2. 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 delete
		'G'	Historical account to keep
2	1	'A'	Line code
3	6	ttttttt	Table number
9	8	DDMMCCYY	Historical account date
19	1		Option
			- when the action code is equal to 'G', storing of the historical account whose date is equal to the date specified. If there is no date, all historical accounts are stored.
			- When the action code is equal to 'S', deletion of the historical account whose date is equal to the date specified.
		'<'	- When the action code is equal to 'G', storing of all historical accounts whose dates are strictly smaller than the date specified.
			- When the action code is equal to 'S', deletion of all historical accounts whose dates are strictly smaller than the date specified.
		'>'	- When the action code is equal to 'G', storing of all historical accounts whose dates are higher than or equal to the date specified
			- When the action code is equal to 'S', deletion of all historical accounts whose dates are higher than or equal to the date specified.

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

For more details, see the Pactables Reference Manual.

10.3. DESCRIPTION OF STEPS

RETA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

REORGANIZATION OF TABLE CONTENTS: PTA400

```
.Permanent input files:
 -Table-description file
  EFN : $NMTU.$TRTABTD
 -Error-message file
  EFN : $NMTU.$TRTABTE
 -Table-contents file
  EFN : $NMTU.$TRTABTV
 -User parameter file
  EFN : $NMTU.$TRTABTG

.Input transaction file:
 -Reorganization requests
  EFN : TMBRETA

.Output file:
 -Reorganized-contents file
  EFN : TPAC7TX
 -Reorganized-table list file
  EFN : TPAC7DE
```

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

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

VALIDIDATION OF TABLE CONTENTS: PTA410

```
.Updating input file :
 -Reorganization requests

.Input file:
 -Reorganized-contents file
  EFN : TPAC7TX

.Output file:
 -Validated-contents file
  EFN : TPAC7TW
```

REORGANIZATION OF TABLE-DESCRIPTIONS: PTA420

```
.Permanent input file:
 -Table-description file
  EFN : $NMTU.TRTABTD

.Input file:
 -Reorganized-table list file
  EFN : TPAC7DE
```

```
.Output files:
 -Reorganized table-description file
  EFN : TPAC7TS
 -Table-description print request
  EFN : TPAC7ML
```

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

```
BUILDING OF BACKUP FILE:  PTA430

.Input files:
 -Validated-contents fille
  EFN : TPAC7TW
 -Reorganized-description file
  EFN : TPAC7TS

.Ouput file:
 -Backup file resulting from
  reorganization
  EFN : TPAC7TC
TG FILE BACKUP:  PTASVG

.Permanent input file:
 -User-parameter file
  EFN : $NMTU.$TRTABTG
.Output file:
 -Table backup
  EFN : TPAC7SG
```

10.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*            TABLE REORGANIZATION                   *';
COMM '*            ====================                   *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '* &SIZEWK : WORK FILE SIZE IN TRACKS          (3)   *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    CTGENY='/G+1',
    RFGEN=&CTGEN$CTBU,
    SVTC='$NMBU.$TRTABTC'&RFGEN'',
    SIZEWK=03;
OVL HOLD;
CR  IF=*RETA,
    OF=(TMBRETA,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTA400 ***';
STEP PTA400,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  100;
     ASG PAC7DR,TMBRETA,TEMPRY,&RFTM,END=PASS;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,NBBUF=1,READLOCK=STAT;
     ASG PAC7TX,TPAC7TX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TX,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TX,NBBUF=1;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM,END=PASS;
     ALC PAC7DE,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7DE,NBBUF=1;
     ASG PAC7IR,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=05,&RFTM);
ESTP;
JUMP ERR,SW10,EQ,1;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA410 ***';
STEP PTA410,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  80;
     ASG PAC7MB,TMBRETA,TEMPRY,&RFTM;
     ASG PAC7TX,TPAC7TX,TEMPRY,&RFTM;
     DEF PAC7TX,NBBUF=1;
     ASG PAC7TW,TPAC7TW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TW,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TW,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA420 ***';
STEP PTA420,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  100;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TS,TPAC7TS,TEMPRY,&RFTM,END=PASS;
```

```
     ALC PAC7TS,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TS,NBBUF=1;
     ASG PAC7DE,TPAC7DE,TEMPRY,&RFTM,END=PASS;
     DEF PAC7DE,NBBUF=1;
     ASG PAC7ML,TPAC7ML,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ML,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7ML,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA430 ***';
STEP PTA430,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  100;
     ASG PAC7TS,TPAC7TS,TEMPRY,&RFTM,END=PASS;
     DEF PAC7TS,NBBUF=1;
     ASG PAC7TW,TPAC7TW,TEMPRY,&RFTM,END=PASS;
     DEF PAC7TW,NBBUF=1;
     ASG PAC7TC,TPAC7TC,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TC,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TC,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTASVG ***';
STEP PTASVG,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU;
     ASG PAC7TC,TPAC7SG,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TC,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TC,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** CREATE TC ***';
CREATE INFILES=((TPAC7TC,TEMPRY,&RFTM)
               (TPAC7SG,TEMPRY,&RFTM)),
       OUTFILE=&SVTC;
JUMP SHFT$CTBU;
SHFTY:
SHIFT $NMBU.$TRTABTC;
SHFTN:
JUMP END;
ERR:
SEND ' PTEXRETA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

11. BACKUP (SVTA)

11.1. INTRODUCTION

TABLE BACKUP (SVTA): INTRODUCTION

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

EXECUTION CONDITION

The files must be closed to on-line use.

USER INPUT

None.

11.2. DESCRIPTION OF STEPS

SVTA: DESCRIPTION OF STEPS

TD BACKUP: PTASVD

```
.Permanent input files:
 -Table-description file
  EFN : $NMTU.$TRTABTD
.Output file:
 -Table backup
  EFN : TPAC7SD
```

TV BACKUP: PTASVV

```
.Permanent input file:
 -Table-contents file
  EFN : $NMTU.$TRTABTV
.Output file:
 -Table backup
  EFN : TPAC7SV
```

TG BACKUP: PTASVG

```
.Permanent input file:
 -User-parameter file
  EFN : $NMTU.$TRTABTG
.Output file:
 -Table backup
  EFN : TPAC7SG
```

11.3. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*                      BACKUP                       *';
COMM '*                      ======                       *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*    SVTD : OUTPUT BACKUP FILE NAME                 *';
COMM '*                              ($NMBU.$TRTABSD/G+1)  *';
COMM '*    SVTG : OUTPUT BACKUP FILE NAME                 *';
COMM '*                              ($NMBU.$TRTABSG/G+1)  *';
COMM '*    SVTV : OUTPUT BACKUP FILE NAME                 *';
COMM '*                              ($NMBU.$TRTABSV/G+1)  *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     RFTM='DVC=$DVTM,MD=$MDTM',
     CTGENY='/G+1',
     RFGEN=&CTGEN$CTBU,
     SIZEWK=05,
     SVTC='$NMBU.$TRTABTC'&RFGEN'';
COMM '*** PTASVD ***';
STEP PTASVD,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU;
     ASG PAC7TC,TPAC7SD,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TC,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TC,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PTASVV ***';
STEP PTASVV,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU;
     ASG PAC7TC,TPAC7SV,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TC,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TC,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PTASVG ***';
STEP PTASVG,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU;
     ASG PAC7TC,TPAC7SG,TEMPRY,&RFTM,END=PASS;
     ALC PAC7TC,SZ=&SIZEWK,UNIT=TRACK,INCRSZ=01;
     DEF PAC7TC,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** CREATE TC ***';
CREATE INFILES=((TPAC7SD,TEMPRY,&RFTM)
               (TPAC7SV,TEMPRY,&RFTM)
               (TPAC7SG,TEMPRY,&RFTM)),
       OUTFILE=&SVTC;
JUMP SHFT$CTBU;
SHFTY:
SHIFT $NMBU.$TRTABTC;
SHFTN:
JUMP END;
ERR:
SEND ' PTEXSVTA - ABNORMAL END OF RUN (I/O ERROR) ';
```

```
   LET  SEV 3;
   END:
```

12. RESTORATION (RSTA)

12.1. INTRODUCTION

RESTORATION (RSTA): INTRODUCTION

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

EXECUTION CONDITION

On-line access 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 abnormal end, the procedure can be restarted as it is, after correcting the problem.

USER INPUT

None.

12.2. DESCRIPTION OF STEPS

RSTA: DESCRIPTION OF STEPS

RESTORATION OF TD: PTARSD

.Permanent output file:
 -Table-description file
  $NMTU.$TRTABTD

.Permanent input file:
 -Table backup
  $NMBU.$TRTABTC

RESTORATION OF TV: PTARSV

.Permanent output file:
 -Table-contents file
  $NMTU.$TRTABTV

.Permanent input file:
 -Table backup
  $NMBU.$TRTABTC

RESTORATION OF TG: PTARSG

.Permanent output file:
 -User parameter file
  $NMTU.$TRTABTG

.Permanent input file:
 -Table backup
  $NMBU.$TRTABTC

12.3. EXECUTION JCL

```
COMM '*******************************************************';
COMM '*                     RESTORATION                     *';
COMM '*                     ===========                     *';
COMM '*                                                     *';
COMM '* SYMBOLICS IN USE :                                  *';
COMM '*    SVTD : INPUT BACKUP FILE NAME ($NMBU.$TRTABSD)   *';
COMM '*    SVTG : INPUT BACKUP FILE NAME ($NMBU.$TRTABSG)   *';
COMM '*    SVTV : INPUT BACKUP FILE NAME ($NMBU.$TRTABSV)   *';
COMM '*                                                     *';
COMM '*******************************************************';
MVL  SVTC='$NMBU.$TRTABTC',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     RFTM='DVC=$DVTM,MD=$MDTM';
COMM '*** ALLOCATION : TD       ***';
IV   PTINALTD ($NMLI.$LIBJCLT,&RFLI);
COMM '*** PTARSD ***';
STEP PTARSD,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  100;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU;
     ASG PAC7TC,&SVTC,&RFBU;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** ALLOCATION : TV       ***';
IV   PTINALTV ($NMLI.$LIBJCLT,&RFLI);
COMM '*** PTARSV ***';
STEP PTARSV,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU;
     ASG PAC7TC,&SVTC,&RFBU;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** ALLOCATION : TG       ***';
IV   PTINALTG ($NMLI.$LIBJCLT,&RFLI);
COMM '*** PTARSG ***';
STEP PTARSG,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU;
     ASG PAC7TC,&SVTC,&RFBU;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXRSTA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

13. LIST OF TABLE DESCRIPTIONS (LDTA)

13.1. INTRODUCTION

LDTA: LIST OF TABLE DESCRIPTIONS

INTRODUCTION

This procedure prints table descriptions.

EXECUTION CONDITION

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

13.2. 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 not taken into account.

13.3. DESCRIPTION OF STEPS

LDTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

TABLE-DESCRIPTION PRINTING: PTA290

```
.Permanent input file:
 -Table-description file
  $NMTU.$TRTABTD

.Input transaction file:
 -Print request
  TMBLDTA

.Output report:
 -Table-description printout
```

13.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*            LIST OF TABLE DESCRIPTIONS             *';
COMM '*            ==========================             *';
COMM '*                                                   *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
CR   IF=*LDTA,
     OF=(TMBLDTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTA290 ***';
STEP PTA290,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TMBLDTA,TEMPRY,&RFTM;
     DEF PAC7MB,NBBUF=1;
     ASG PAC7ID,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXLDTA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

14. PARAMETER UPDATE (PMTA)

14.1. INTRODUCTION

PMTA: USER PARAMETER UPDATE

INTRODUCTION

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

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 material in use allows Batch/TP concurrency.

14.2. 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	tttttt	Table code
18	3	nnn	Line number
21	60		Access authorizations: 20 access authorizations may be entered in this field, with, for each authorization:
	1	n	pos. 1: sub-schema number ('*' for all sub-schemas)
	1	n	pos. 2: sub-system number ('*' for all sub-systems)
	1	x	pos. 3: autorization (0,1 or 2)

'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 this code is not authorized.

14.3. DESCRIPTION OF STEPS

PMTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

PARAMETER UPDATE: PTA100

```
.Permanent input files:
 -Table-description file
  $NMTU.$TRTABTD
 -Error-message file
  $NMTU.$TRTABTE

.Permanent input-output file:
 -User parameter file
  $NMTU.$TRTABTG

.Input transaction file:
 -Extraction requests
  TMBPMTA

.Output file:
 -Parameter printing requests
  TPAC7NU

.Output report:
 -Printing of descriptions
```

PRINTING OF USER PARAMETERS: PTA120

```
.Permanent input files:
 -Table description file
  $NMTU.$TRTABTD
 -User parameter file
  $NMTU.$TRTABTG

.Input transaction file:
 -Print requests
  TPAC7NU

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

14.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*            UPDATE OF USER PARAMETERS              *';
COMM '*            =========================              *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '* &SIZENU : WORK FILE SIZE IN TRACKS          (10)  *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    SIZENU=10;
OVL HOLD;
CR  IF=*PMTA,
    OF=(TMBPMTA,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTA100 ***';
STEP PTA100,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU;
     ASG PAC7MV,TMBPMTA,TEMPRY,&RFTM;
     ASG PAC7NU,TPAC7NU,TEMPRY,&RFTM,END=PASS;
     ALC PAC7NU,SZ=&SIZENU,UNIT=TRACK,INCRSZ=01;
     DEF PAC7NU,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA120 ***';
STEP PTA120,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7NU,TPAC7NU,TEMPRY,&RFTM;
     DEF PAC7NU,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXPMTA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

15. TABLE EXTRACTION (EXTA)

15.1. 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.

15.2. 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	tttttt	Table number
9	8	DDMMCCYY	Historical account date
17	1		Not used
18	1		Sub-system selection
		blank	No sub-system selection
		1 TO 0	Number of selected sub-system
19	1		Data delimiter
		blank	'/'

15.3. DESCRIPTION OF STEPS

EXTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

EXTRACTION OF TABLE DATA: PTA150

```
.Permanent input files:
 -Table-description file
  $NMTU.$TRTABTD
 -Error message file
  $NMTU.$TRTABTE
 -Table contents file
  $NMTU.$TRTABTV
 -User parameter file
  $NMTU.$TRTABTG

.Input transaction file:
 -Extraction requests
  TMBEXTA

.Output file:
 -Extracted transactions
  TPAC7EX

.Output report:
 -Transaction review
```

PRINTING OF EXTRACTED TRANSACTIONS: PTA160

```
.Permanent input file:
 -Table description file
  $NMTU.$TRTABTD

.Input transaction file:
 -Extracted transactions
  TPAC7EX

.Output report:
 -Printing of extracted data

.Output file:
 -Extracted transactions
  TPAC7NU

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

15.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*               TABLE EXTRACTION                    *';
COMM '*               ================                    *';
COMM '*                                                   *';
COMM '* PARAMETERS IN USE :                               *';
COMM '*                                                   *';
COMM '*    &USER   : USER CODE FOR USER EXTRACTED         *';
COMM '*              TRANSACTIONS FILE SUFFIX             *';
COMM '*    &SZWK   : SORTWORK FILE SIZE                   *';
COMM '*                                                   *';
COMM '* OUTPUT :                                          *';
COMM '*                                                   *';
COMM '*    OUTPUT TRANSACTIONS ARE STORED IN THE          *';
COMM '*    LIBRARY $NMLI.$LIBSUT                          *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    USER='$USER',SZWK=3;
OVL HOLD;
CR   IF=*EXTA,
     OF=(TMBEXTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP PTA150,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7MV,TMBEXTA,TEMPRY,&RFTM;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=2,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7EX,TPAC7EX,TEMPRY,&RFTM,END=PASS;
     DEF PAC7EX,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTA160 ***';
STEP PTA160,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7EX,TPAC7EX,TEMPRY,&RFTM;
     DEF PAC7EX,NBBUF=1;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7NU,TPAC7NU,TEMPRY,&RFTM,END=PASS;
     DEF PAC7NU,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=&SZWK,&RFTM);
ESTP;
LMN  SL INFILE=(TPAC7NU,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSUT,&RFLI),
     COM='MV INFILE:MBUPTA_EXTA'&USER',TYPE=DAT,INFORM=SARF,
          NUMBER=(1,1),REPLACE;';
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTUSEXTA - ABNORMAL END OF RUN (I/O ERROR) ';
```

```
LET  SEV 3;
END:
```

16. DIRECT CONSULTATION OF TABLES (TUTA)

16.1. INTRODUCTION

TUTA: DIRECT CONSULTATION OF TABLES

INTRODUCTION

The TUTA procedure extracts tables in the form of tables without historical account 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.

16.2. 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.

16.3. DESCRIPTION OF STEPS

TUTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

DIRECT CONSULTATION OF TABLES: PTAU80

```
.Permanent input files:
 -Table-description file
  $NMTU.$TRTABTD
 -Error-message file
  $NMTU.$TRTABTE
 -Table-contents file
  $NMTU.$TRTABTV
 -User-parameter file
  $NMTU.$TRTABTG

.Input transaction file:
 -Request transactions
  TMBTUTA

.Permanent output files:
 -Table-description file
  $NMTU.$TRTABAD
 -Table-contents file
  $NMTU.$TRTABAV

.Output report:
 -Transaction report
```

16.4. EXECUTION JCL

```
COMM '*****************************************************';
COMM '*                                                   *';
COMM '*            DIRECT CONSULTATION OF TABLES          *';
COMM '*            ============================           *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
CR   IF=*TUTA,
     OF=(TMBTUTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION DES FICHIERS: AD ET AV ***';
IV   PTINALAD ($NMLI.$LIBJCLT,&RFLI);
IV   PTINALAV ($NMLI.$LIBJCLT,&RFLI);
COMM '*** PTAU80 ***';
STEP PTAU80,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ 100;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7MX,TMBTUTA,TEMPRY,&RFTM;
     ASG PAC7AD,$NMTU.$TRTABAD,&RFTU;
     ASG PAC7AV,$NMTU.$TRTABAV,&RFTU;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=05,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXTUTA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

17. DISPATCHED TABLE MANAGEMENT (DTM OPTION)

17.1. 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 as input in the CDT2 procedure in order to update the 'outdated' version of the table description.

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, which are called 'slave' files. These files must therefore remain closed to on-line use.

17.2. 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 table descriptions are compared.

17.3. DESCRIPTION OF STEPS (CDT1)

CDT1: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

CHECK OF TRANSACTIONS: PTAD05

```
.Permanent input files:
 -'Master' table-description file
   &TDMAST
 -Error-message file
  $NMTU.$TRTABTE
 -User-parameter file
  $NMTU.$TRTABTG

.Input transaction file:
 -Comparison request transactions
  TMBCDT1

.Output file:
 -Validated comparison request transactions
  TPAC7MX

.Output report:
 -Transaction report
```

TABLE-DESCRIPTION COMPARISON AND EXTRACTION: PTAD10

```
.Permanent input files:
 -'Master' table-description file
   &TDMAST
 -Error message file
  $NMTU.$TRTABTE
 -'Slave' table-description file
   &TDSLAV

.Input transaction file:
 -Validated transactions
  TPAC7MX

.Output file:
 -Comparison result to be used as input of
  the CDT2 procedure
  TPAC7TX

.Output report:
 -Extraction printout
```

17.4. EXECUTION JCL (CDT1)

```
COMM '*****************************************************';
COMM '*         TABLE DESCRIPTION COMPARISON              *';
COMM '*         ============================              *';
COMM '*                                                   *';
COMM '* PARAMETERS IN USE :                               *';
COMM '*                                                   *';
COMM '*    &USER   : USER CODE FOR USER EXTRACTED         *';
COMM '*              TRANSACTIONS FILE SUFFIX             *';
COMM '*    &TDMAST : MASTER TABLE DESCRIPTION FILE        *';
COMM '*    &TDSLAV : SLAVE  TABLE DESCRIPTION FILE        *';
COMM '*                                                   *';
COMM '* OUTPUT :                                          *';
COMM '*                                                   *';
COMM '*    OUTPUT TRANSACTIONS ARE STORED IN THE          *';
COMM '*    LIBRARY $NMLI.$LIBSUT                          *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',USER='$USER',
    TDMAST=,
    TDSLAV=;
OVL HOLD;
CR   IF=*CDT1,
     OF=(TMBCDT1,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTAD05 ***';
STEP PTAD05,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7MV,TMBCDT1,TEMPRY,&RFTM;
     ASG PAC7TD,&TDMAST,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7MX,TPAC7MX,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MX,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTAD10 ***';
STEP PTAD10,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,&TDMAST,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7TS,&TDSLAV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TS,NBBUF=1,READLOCK=STAT;
     ASG PAC7MX,TPAC7MX,TEMPRY,&RFTM;
     DEF PAC7MX,NBBUF=1;
     ASG PAC7TX,TPAC7TX,TEMPRY,&RFTM,END=PASS;
     DEF PAC7TX,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
LMN  SL INFILE=(TPAC7TX,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSUT,&RFLI),
     COM='MV INFILE:MBCDT2_CDT1'&USER',TYPE=DAT,INFORM=SARF,
```

```
          NUMBER=(1,1),REPLACE;';
   JUMP ERR,SW20,EQ,1;
   JUMP END;
   ERR:
   SEND ' PTEXCDT1 - ABNORMAL END OF RUN (I/O ERROR) ';
   LET  SEV 3;
   END:
```

17.5. DESCRIPTION OF STEPS (CDT2)

CDT2: DESCRIPTION OF STEPS

UPDATE OF 'SLAVE' FILES, TABLE-DESCRIPTIONS AND
RECOGNITION OF THE FILE EXTRACTED BY CDT1: PTAD20

```
.Input files:
 -'Slave' file of table-descriptions
  &TDSLAV
 -Error-message file
  $NMTU.$TRTABTE

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

.Input transaction file:
 -Result extracted from comparison in the
  CDT1 procedure
  TPAC7TX

.Output report:
 -Update report
```

17.6. EXECUTION JCL (CDT2)

```
COMM '*****************************************************';
COMM '* TABLE DESCRIPTION AND CONTENTS SLAVE FILES UPDATE  *';
COMM '* ==============================================  *';
COMM '*                                                   *';
COMM '* PARAMETERS :                                     *';
COMM '*                                                   *';
COMM '*    &USER   : USER CODE USED AS SUFFIX FOR         *';
COMM '*              EXTRACTED TRANSACTIONS INPUT FILE    *';
COMM '*    &TDSLAV : TABLE DESCRIPTION SLAVE FILE         *';
COMM '*    &TVSLAV : TABLE CONTENTS FILE LINKED TO TABLE  *';
COMM '*              DESCRIPTION SLAVE FILE               *';
COMM '*    &SZWK   : SORT FILE SIZE                       *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL USER='$USER',SZWK=5,
    TDSLAV=,
    TVSLAV=,
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
CR   IF=($NMLI.$LIBSUT,&RFLI,SUBFILE=MBCDT2_CDT1&USER),
     OF=(TPAC7TX,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=240,RECFORM=FB),
     COMFILE=($NMLI.$LIBJCLT,&RFLI,SUBFILE=PTEXPDS1),START=2;
COMM '*** PTAD20 ***';
STEP PTAD20,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,&TDSLAV,&RFTU;
     ASG PAC7TV,&TVSLAV,&RFTU;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7TX,TPAC7TX,TEMPRY,&RFTM;
     DEF PAC7TX,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=&SZWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXCDT2 - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

17.7. TABLE CONTENTS UPDATE (CVTA)

CVTA: COMPARISON AND UPDATING OF TABLE CONTENTS

INTRODUCTION

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

EXECUTION CONDITION

This procedure reads the Pactables files. It can be executed even if the files are open to on-line use.

17.8. 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	tttttt	Table number
9	8	DDMMCCYY	Update date: beginning
17	4		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 table contents to which the user ('*'-line) has access can be extracted.

17.9. DESCRIPTION OF STEPS (CVTA)

CVTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

TABLE-CONTENTS COMPARISON: PTAV10

```
.Permanent input files:
-Table-description file
 $NMTU.$TRTABTD
-Error-message file
 $NMTU.$TRTABTE
-Table-contents file
 $NMTU.$TRTABTV
-User-parameter file
 $NMTU.$TRTABTG

.Input transaction file:
-Comparison requests
 TMBCVTA

.Output file:
-Comparison result
 TPAC7EX

.Output report:
 -Transaction report
```

EXTRACTION OF UPDATE TRANSACTIONS: PTAV20

```
.Permanent input file:
-Table-Description file
 $NMTU.$TRTABTD

.Input transaction file:
-Comparison result
 TPAC7EX

.Output file:
-Update transactions for use as
 input of UPTA)
 TPAC7NU

.Output report:
 -Printing of extracted transactions
```

17.10. EXECUTION JCL (CVTA)

```
COMM '*****************************************************';
COMM '*          CVTA - TABLE CONTENTS COMPARISON         *';
COMM '*          ================================         *';
COMM '*                                                   *';
COMM '* PARAMETERS :                                      *';
COMM '*                                                   *';
COMM '*    &USER : USER CODE USED AS SUFFIX FOR           *';
COMM '*            UPDATE TRANSACTIONS OUTPUT FILE        *';
COMM '*    &TD   : TABLE DESCRIPTION FILE                 *';
COMM '*    &TV   : TABLE CONTENTS FILE LINKED TO TABLE    *';
COMM '*            DESCRIPTION FILE                       *';
COMM '*    &SZEX : COMPARISON OUTPUT FILE SIZE            *';
COMM '*    &SZWK : WORK FILE SIZE                         *';
COMM '*                                                   *';
COMM '* OUTPUT :                                         *';
COMM '*                                                   *';
COMM '*    OUTPUT TRANSACTIONS ARE STORED IN THE          *';
COMM '*    LIBRARY $NMLI.$LIBSUT                          *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL USER='$USER',SZEX=10,SZWK=10,
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
CR   IF=*CVTA,
     OF=(TMBCVTA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTAV10 ***';
STEP PTAV10,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7MV,TMBCVTA,TEMPRY,&RFTM;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7TV,$NMTU.$TRTABTV,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TV,NBBUF=1,READLOCK=STAT;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TG,NBBUF=1,READLOCK=STAT;
     ASG PAC7TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TE,READLOCK=STAT;
     ASG PAC7EX,TPAC7EX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7EX,SZ=&SZEX,UNIT=TRACK,INCRSZ=01;
     DEF PAC7EX,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTAV20 ***';
STEP PTAV20,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7EX,TPAC7EX,TEMPRY,&RFTM;
     DEF PAC7EX,NBBUF=1;
     ASG PAC7TD,$NMTU.$TRTABTD,&RFTU,
         SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7NU,TPAC7NU,TEMPRY,&RFTM,END=PASS;
     DEF PAC7NU,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=&SZWK,&RFTM);
ESTP;
LMN  SL INFILE=(TPAC7NU,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSUT,&RFLI),
```

```
     COM='MV INFILE:MBUPTA_CVTA'&USER',TYPE=DAT,INFORM=SARF,
          NUMBER=(1,1),REPLACE;';
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PTEXCVTA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

18. TABLE RETRIEVAL FROM RELEASES 7.x (RxTA) to 1.2

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

RETRIEVAL FROM RELEASE 7.3 (R3TA)

The retrieval of existing tables and files, which allows Tables of the 7.3 release to be used in the 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.

.ote : 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.

```
***********
* WARNING *
***********
```

DO NOT REPLACE PTACCE 2.5 by PTACCE 7.3 used for retrieval 7.3 -> 2.5 ; Load, in the CU BATCH library (environment 7.3), the following programs : PTACCE PTAXVD PTAXVV PTAXVG contained in the CUT20 culib. Linker PTAXVD PTAXVV PTAXVG . In the Pactables 2.5 R3TA procedure, replace the reference to the Pactables 2.5 LMLIB by the Pactables 7.3 LMLIB for the programs mentioned above.

18.2. USER INPUT

USER INPUT

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

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

18.3. DESCRIPTION OF STEPS (R3TA)

R3TA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

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
&TD73

.Output file:
-Backed up 7.3 file
TPAC7TD

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
&TV73

.Output file:
-7.3 backed up file
TPAC7TV

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
&TG73

.Output file:
-7.3 backed up file
TPAC7TG

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
TPAC7TC
-User-parameter file

.Output file:
-Temporary 2.5 backup : PAC7TR
$NMBU.$TRTABTC

.Output report:
-Retrieval report

18.4. EXECUTION JCL (R3TA)

```
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    TD73='OLDTD',
    TV73='OLDTV',
    TG73='OLDTG',
    CTGENY='/G+1',
    RFGEN=&CTGEN$CTBU,
    SVTC='$NMBU.$TRTABTC'&RFGEN'';
OVL HOLD;
CR  IF=*R3TA,
    OF=(TMBR3TA,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PTAXVD ***';
STEP PTAXVD,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,&TD73;
     ASG PAC7TC,TPAC7TD,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PTAXVV ***';
STEP PTAXVV,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TV,&TV73;
     ASG PAC7TC,TPAC7TV,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PTAXVG ***';
STEP PTAXVG,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TG,&TG73;
     ASG PAC7TC,TPAC7TG,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** CREATE OF TC 1.2 *';
CR  INFILES=((TPAC7TD,TEMPRY,&RFTM)
       (TPAC7TV,TEMPRY,&RFTM)
       (TPAC7TG,TEMPRY,&RFTM)),
    OF=(TPAC7TC,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=$CISEQ,RECSZ=1057,RECFORM=V);
COMM '*** PTAR20 ***';
STEP PTAR20,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  100;
     ASG PAC7TC,TPAC7TC,TEMPRY,&RFTM;
     ASG PAC7TR,&SVTC,&RFBU;
     ASG PAC7MB,TMBR3TA,TEMPRY,&RFTM;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP SHFT$CTBU;
SHFTY:
SHIFT $NMBU.$TRTABTC;
SHFTN:
JUMP END;
ERR:
SEND ' PTEXR3TA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

18.5. 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.

.OTE : 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.

18.6. DESCRIPTION OF STEPS (R2TA)

R2TA: DESCRIPTION OF STEPS

INPUT RECOGNITION: CREATE

CONVERSION OF THE TG FILE FROM 7.2 TO 7.3: PTARTG

.Input file:
-Sequential 7.2 TG file
TPAC7TG

.Output file:
-7.3 TG file
$NMTU.$TRTABTG
-Conversion report

.Input-output file:
-7.2 TD file
&TD72

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
&TD72

.Output file:
-7.3 backup file
TPAC7TD

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
&TV72

.Output file:
-7.3 backup file
TPAC7TV

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
&TG72

.Output file:
-7.3 backup file
TPAC7TG

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
TPAC7TC
-User-parameter file

.Output file:
-Temporary 2.5 backup : PAC7TR
$NMBU.$TRTABTC

.Output report:
-Conversion report

18.7. EXECUTION JCL (R2TA)

```
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    TD72='OLDTD',
    TV72='OLDTV',
    TG72='OLDTG',
    CTGENY='/G+1',
    RFGEN=&CTGEN$CTBU,
    SVTC='$NMBU.$TRTABTC'&RFGEN'';
OVL HOLD;
CR  IF=*R2TA,
    OF=(TMBR2TA,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** TG FILE ALLOCATION ***';
IV  PTINALTG, ($NMLI.$LIBJCLT,&RFLI);
COMM '*** COPY OF TG 7.2 ***';
CR  IF=(&TG72,&RFTU),
    OF=(TPAC7TG,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80);
COMM '*** PTARTG ***';
STEP PTARTG,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,&TD72;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU;
     ASG PAC7AG,TPAC7TG,TEMPRY,&RFTM;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTAXVD ***';
STEP PTAXVD,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TD,&TD72;
     ASG PAC7TC,TPAC7TD,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PTAXVV ***';
STEP PTAXVV,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TV,&TV72;
     ASG PAC7TC,TPAC7TV,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PTAXVG ***';
STEP PTAXVG,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7TG,$NMTU.$TRTABTG,&RFTU;
     ASG PAC7TC,TPAC7TG,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** CREATE OF TC 1.2 *';
CR  INFILES=((TPAC7TD,TEMPRY,&RFTM)
       (TPAC7TV,TEMPRY,&RFTM)
       (TPAC7TG,TEMPRY,&RFTM)),
    OF=(TPAC7TC,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=$CISEQ,RECSZ=1057,RECFORM=V);
COMM '*** PTAR20 ***';
STEP PTAR20,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA;
     SZ  100;
     ASG PAC7TC,TPAC7TC,TEMPRY,&RFTM;
     ASG PAC7TR,&SVTC,&RFBU;
     ASG PAC7MB,TMBR2TA,TEMPRY,&RFTM;
```

```
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP SHFT$CTBU;
SHFTY:
SHIFT $NMBU.$TRTABTC;
SHFTN:
JUMP END;
ERR:
SEND ' PTEXR2TA - ABNORMAL END OF RUN (I/O ERROR) ';
LET  SEV 3;
END:
```

19. COMPATIBILITY BETWEEN PACTABLES 2.5 AND VA PAC 1.6

19.1. COMPATIBILITY BETWEEN PACTABLES 2.5 AND VA PAC 1.6

COMPATIBILITY BETWEEN PACTABLES 2.5 and VA Pac 1.6

If you use Pactables 2.5 and generate table descriptions of VA Pac from version 1.6 and higher versions (or from a former version), you need using 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 VA Pac as they are not compatible with Pactables 2.5.

20. INSTALLATION

20.1. INTRODUCTION

INTRODUCTION

The installation procedure is executed in three main steps:

. Preparation for installation,
. Installation,
. On-line and batch tests.

Pactables installation is totally independent of the presence of the VA Pac system on site. So Pactables can be a separated system.

Before proceeding with the installation, the user must be familiar with the technical characteristics of the Pactables function described in this manual. This information is necessary to prepare the environment required for the installation procedure.

Once the environment is prepared, the installation can be started. It comprises the following steps:

PREPARATION

. Backup of the installation tape,

. Allocation of a JCL library,

. Unloading of the Pactables complete operations and installation JCL,

. Adaptation of the JCL to the site's specific needs.

INSTALLATION

1. Preallocation of system files,
2. TDS preparation,
3. File installation,
4. TDS generation,
5. Program link-edit,
6. Installation of a Pactables 2.5 test database,
7. User paramater update,
8. TDS start-up.

TESTS

. On-line tests,

. Tests on batch procedures.

IMPORTANT NOTE

For a CATALOGUE-type installation, it is COMPULSORY to create first the slave catalogue and the directories required by the installation.

The 'AUTOATTACH' option must be specified for the installation catalogue.

20.2. INSTALLATION TAPE

INSTALLATION TAPE

The installation tape (1600 BPI, standard labels) includes the following files:

Rank	Label	Content
1	SVF.JCLT	Skeleton JCL for installation and operations in French
2	SVE.JCLT	Skeleton JCL for installation and operations in English
3	SVF.TC	backup of French tables (content, description and parameters)
4	SVE.TC	backup of English tables (content, description and parameters)
5	SV.SLT	sources of the TDS and the first TPR
6	SV.CUBT	batch compile-units
7	SV.CUTT	TPR compile-units
8	SV.TE	Pactables labels' system files
9	SV.CUT20	century-management COMPILE-UNITs

20.3. JCL INSTALLATION

INSTALLATION OF THE COMPLETE JCL

This installation comprizes three steps:

1- Allocation of a JCL library: '$NMLI.$LIBJCLT'.

If VisualAge Pacbase 2.0 is already installed on the site and if you want to integrate Pactables into it, you can use the name of the VisualAge Pacbase JCL library IF the VA Pac JCL has already undergone the parameter interpretation procedure. In this case, you need not perform the allocation. In any other cases, the name of the JCL library must conform to the value taken by the parameters which make up its parameterized name.

```
Command (JCL):
LIBALLOC SL,($NMLI.$LIBJCLT,DVC=...,MD=...,SIZE=03),
MEMBERS=50,COMPACT;
```

2- Loading of this library from the first file (SV.JCLT) by LIBMAINT.

```
Command (JCL):LIBMAINT SL,INFILE=(SV.JCLT,DVC=MT/T9,MD=PACxxx,
FSN=1),LIB=($NMLI.$LIBJCLT,DVC=...,MD=...),
COMMAND='MOVE INFILE:*,REPLACE;';
```

3- Adaptation of the JCL to the site's specificities.

You adapt the JCL by modifying, under the editor, the 'PTZZVALS' member of this library (the default value of each parameter is substituted by the value specific to the site). Then you must run the 'PTZZEXEC' procedure which prepares the parameters' replacement JCL (EXEC PTZZEXEC VL=PTZZJCL BRIEF;), and submit this JCL (SUBMIT PTZZJCL;).

NOTES :

When running 'PTZZEXEC', error messages such as 'SUBSTITUTION FAILED' are displayed. They do not hinder the execution process.

Pactables TDS is usually linked in the Pactables batch load-module library. So it is important to give $NMTD a value which differs from that of Pactables standard load-modules, in particular 'PACB' (see the list of batch load-modules in chapter 'INTRODUCTION').

If you want to run VisualAge Pacbase and Pactables under the same TDS, you must give the $LIBCUBT , $LIBCUTT, $LIBLMT and $LIBSLT parameters the same values as those chosen at the time of VA Pac installation, so that the Pactables elements are directly unloaded in the associated libraries.

PACTABLES JCL MEMBERS

All these members are in the '$NMLI.$LIBJCLT' library.

OPERATIONS PROCEDURES

Member	Content	Nature
PTEXCDT1	Description comparison	JCL
PTEXCDT2	Description updating	JCL
PTEXCVTA	Content comparison	JCL
PTEXGETT	Table generation	JCL
PTEXINTA	Pactables initialization	JCL
PTEXJOBL	JOB submission	JCL
PTEXLDTA	List of table descriptions	JCL
PTEXPDSL	Technological members to transform an IOF member into UFAS 80 characters	JCL
PTEXPMTA	Parameter update	JCL
PTEXRETA	Table reorganization	JCL
PTEXRSTA	TD, TV, TG files restoration	JCL
PTEXR2TA	Retrieval of tables 7.2 --> 2.5	JCL
PTEXR3TA	Retrieval of tables 7.3, 8.xx --> 2.5	JCL
PTEXSVTA	Backup of TD, TV, TG files	JCL
PTEXTDPT	Pactables TDS submission	JCL
PTEXTUTA	Optimized production turnover	JCL

USER PROCEDURES

Member	Content	Nature
PTUSEXTA	Table extraction	JCL
PTUSIMTA	Table import	JCL
PTUSPRTA	Table printing	JCL
PTUSUPTA	Table updating	JCL

INSTALLATION PROCEDURES

Member	Content	Nature
PTINAL*	Allocation of current Pactables files	JCL
PTINALLI	Library allocation	JCL
PTINBLNK	Link of batch programs in LMLIB	JCL
PTINBLSO	Model linker for batch programs	JCL
PTININTJ	TJ file initialization	JCL
PTINMGEN	Pactables TDS generation	JCL
PTINMPRE	Pactables TDS preparation	JCL
PTINPRPT	Pactables file allocation	JCL
PTINRST1	Restoration of a test database	JCL
PTINTLNK	Link of on-line programs in SMLIB	JCL
PTINTLSO	Model linker for on-line programs	JCL
PTINUNLD	Installation of files and programs	JCL
PTZZEDIT	JCL parameters	JCL
PTZZEXEC	JCL parameters	JCL
PTZZJCL	JCL parameters	JCL
PTZZVALS	Default parameters	DAT

* : takes the suffix value of all Pactables files

example : PTINALTD is the allocation JCL of the table-description file.

CALL OF PROCEDURES

- These members contain the user input and are located in the '$NMLI.$LIBINVT' library.
  They call by INVOKE the procedures of the '$NMLI.$LIBJCLT' library.

Member	Content	Nature
PTIVCDT1	Description comparison	JCL
PTIVCDT2	Description update	JCL
PTIVCVTA	Data comparison	JCL
PTIVEXTA	Table extraction	JCL
PTIVGETT	Table generation	JCL
PTIVIMTA	Table import	JCL
PTIVINTA	Table initialization	JCL
PTIVLDTA	List of tables	JCL
PTIVPMTA	Parameter updating	JCL
PTIVPRTA	Table printing	JCL
PTIVRETA	Table reorganization	JCL
PTIVRSTA	Table restoration	JCL
PTIVR2TA	Retrieval of tables 7.2	JCL
PTIVR3TA	Retrieval of tables 7.3, 8.xx and 1.2	JCL
PTIVSVTA	Table backup	JCL
PTIVTUTA	Optimized production turnover	JCL
PTIVUPTA	Table update	JCL

```
" *********************************************
" *                                           *
" * PTZZEDIT : THIS EDIT PROCEDURE IS CALLED BY *
" * THE JCL INTERPRETER (PTZZEXEC).           *
" * ITS PURPOSE IS TO PREPARE THE USER SUBFILE  *
" * PTZZVALS FOR JCL INTERPRETATION.          *
" *                                           *
" *********************************************
"  WEAKEN RETURN CODES
YW
"  LOADING PTZZVALS
RPTZZVALS
"  REMOVE ALL BLANKS IN THE LINES
^,$S/ //
"  DELETE LINES WHICH DO NOT BEGIN WITH $
VD/^°C$/
" INSERT STRING "^,$S=°C" AT THE BEGINNING OF EACH LINE
^,$S/^/^,$S=°C°°CC/
" INSERT STRING "°C" IN FRONT OFF EACH CARACT "&"
GS/°C&/°C°°CC°C&/
" INSERT STRING "="     ET THE END OF EACH LINE
^,$S/$/=/
```

```
COMM 'VA-PACTABLE 2.5     ';
COMM ****************************************;
COMM *                                      *;
COMM * PTZZEXEC : JCL INTERPRETATION EXEC PROC. *;
COMM * THIS PROCEDURE PREPARES THE USER SUBFILE *;
COMM * PTZZVALS WITH THE PROCEDURE PTZZEDIT.    *;
COMM * THEN IT REPLACES PACTABLE PARAMETERS BY  *;
COMM * USER VALUES IN TARGET SUBFILES SPECIFIED *;
COMM * IN PARAMETER 1, ACCORDING TO THE NAMING  *;
COMM * CONVENTIONS OF LIBRARY SUBFILES.         *;
COMM * EX : EXEC PTZZEXEC VL=PTZZJCL BRIEF      *;
COMM *                                      *;
COMM ****************************************;
ED;
YB
B1
RPTZZEDIT
B0
°E1
Z(JCL)PTZZTEMP
Q
STATUS RESET;
ED  LIB:&1;
YB
B1
RPTZZTEMP
B0
R &0
°E1
Z &0
Q
STATUS RESET;
DELETE PTZZTEMP;
```

```
COMM 'VA-PACTABLE 2.5     ';
COMM '*********************************************';
COMM '*                                           *';
COMM '*  PTZZJCL : THIS PROCEDURE MUST BE EXECUTED    *';
COMM '*  FOR THE INTERPRETATION OF PACTABLE JCL.      *';
COMM '*  BEFORE EXECUTION, PTZZJCL ITSELF HAS TO BE   *';
COMM '*  INTERPRETATED BY THE EXEC PROCEDURE PTZZEXEC *';
COMM '*  EX : EXEC PTZZEXEC VL=PTZZJCL BRIEF;         *';
COMM '*                                           *';
COMM '*********************************************';
VL   ^PTZZ*,
     DV4='DVC=$DVLI',MD4='MD=$MDLI';
LMN  SL LIB=($NMLI.$LIBJCLT,&RFLI),
        COMFILE=*PT80A,PRTFILE=DUMMY;
$IN  PT80A PRINT JVL;
EXEC PTZZEXEC VL=&1;
$EIN PT80A;
LMN  SL LIB=($NMLI.$LIBJCLT,&RFLI),
        COMFILE=*PT80B;
$IN  PT80B PRINT JVL;
PR   LIB:PTZZVALS;
PR   LIB:&1;
$EIN PT80B;
```

```
COMM 'VA-PACTABLE 2.5     ';
*****************************************
*                                       *
*        INSTALLATION PARAMETERS        *
*                                       *
*  REPLACE, IF NEEDED, THE DEFAULT      *
*  VALUE OF EACH PARAMETER.             *
*  EACH PARAMETER LINE IS FORMATTED AS  *
*  FOLLOWS:                             *
*    $NNNNN = VALUE                     *
*                                       *
*                                       *
*  - THE EQUAL SIGN (DELIMITER) AND     *
*    THE AMPERSAND CANNOT BE USED IN    *
*    A PARAMETER VALUE.                 *
*                                       *
*  SUBSEQUENTLY,                        *
*                                       *
*  - ALL LINES WHOSE FIRST NON-BLANK    *
*    CHARACTER IS NOT A DOLLAR SIGN     *
*    ARE CONSIDERED AS COMMENTS.        *
*                                       *
*  - THE EQUAL SIGN (DELIMITER)         *
*    CANNOT BE USED IN A PARAMETER      *
*    VALUE.                             *
*                                       *
*****************************************

******** LANGAGE CODE                   *
$LANG   =  E
******** DEFAULT PACTABLE USER NAME     *
$USER   =  CGI
******** INSTALLATION CATALOG NAME      *
$CTNM   =  PT
******** TDS PACTABLE NAME              *
$NMTD   =  TDST
******** NAME OF INSTALLATION TAPE      *
$TAPE   =  XXXXXX
******** DEVICE OF INSTALLATION TAPE      *
$DVTP   =  CT/M5
************ SUFFIX OF LIBRARIES **********
******** CU BATCH
$LIBCUBT  =  CUBLIBT
******** CU TP
$LIBCUTT  =  CUTLIBT
******** PERMANENT CU
$LIBCUPT  =  CUPLIBT
******** CU OF MODULES CALLING SUB-PROGRAMS GENERATE WITH
******** SIECLE OPTION - MUST BE DIFFERENT OF $LIBCUBT & $LIBCUTT
$LIBCU20  =  CU20LIB
******** JCL
$LIBJCLT  =  JCLLIBT
******** CALL OF JCL PROCEDURE
$LIBINVT  =  INVLIBT
******** PRINT OF JCL IN REPORTS
**       $LIST = SPACE  NO PRINT
**       $LIST = &LIST     PRINT
$LIST     =  &LIST
******* LM
$LIBLMT   =  LMLIBT
******* SM     (SEE NOTE)
$LIBSMT   =  SMLIB
******* USER SL
$LIBSUT   =  SULIBT
******* TDS SL (SEE NOTE)
$LIBSLT   =  SLLIB
******** RADICAL-IDENTIFIER OF PACTABLE FILES **
$TRTAB  =  PT200
******** PREFIX OF PACBASE BATCH USER FILES
** THE VALUE OF $TRPAC MUST BE EQUAL TO THAT OF $NMBU OF PACBASE
$TRPAC  =  PT.PB200.BU
*******  CISIZE OF BATCH FILES
*******  DEFAULT VALUE FOR DISC MS/D500
```

```
$CISEQ  =  14336
*******  MEDIA TYPE OF SEQUENTIAL FILE TC  *****
*** IF CATALOGED FILE
*** VALUES OF $MDSV : T FOR TAPE OR D FOR DISK
$MDSV   =  D
*************** REFERENCES **************
*NMXX   =  PREFIX FILE
*DVXX   =  DEVICE FILE
*MDXX   =  MEDIA  FILE
*CTXX   =  CATALOG (Y OR N)
******** TDS  FILES
$DVTD   =  MS/D500
$MDTD   =  DISC01
$CTTD   =  Y
******** TDS USER FILES
$NMTU   =  PTTU
$DVTU   =  MS/D500
$MDTU   =  DISC02
$CTTU   =  Y
******** BATCH USER FILES
$NMBU   =  PTBU
$DVBU   =  MS/D500
$MDBU   =  DISC03
$CTBU   =  Y
******** TEMPORARY FILES
$DVTM   =  MS/D500
$MDTM   =  DISC04
******** LIBRARIES
$NMLI   =  PTLI
$DVLI   =  MS/D500
$MDLI   =  DISC05
$CTLI   =  Y
************* LINK ENVIRONMENT (SEE NOTE) *
******** NAME OF PACTABLE LINK TPR0       *
$TPR0   =  TPR
*****************************************
```

20.4. INSTALLATION PROCESS

INSTALLATION PROCESS

Once the JCLs are obtained, the Pactables system can be installed via the following operations:

1. System file allocation,
2. Preparation of Pactables TDS,
3. Unloading of files and programs,
4. Generation of Pactables TDS,
5. Link-edit of batch programs and link-edit of the TPRs in the SMLIB prepared for Pactables,
6. Restoration of a Pactables 2.5 test database,
7. User parameter update,
8. TDS submission.

1. ALLOCATION OF SYSTEM FILES
(see the JCL at the end of this subchapter)

System files are allocated by the execution of the 'PTINPRPT' member located in the '$NMLI.$LIBJCLT' library.

This member is a sequence of PREALLOC and LIBALLOC, and can be logically partitioned in the following way:

```
. Allocation of UFAS table files:
          $NMTU.$TRTABTD
          $NMTU.$TRTABTE
          $NMTU.$TRTABTV
          $NMTU.$TRTABTG
          $NMTU.$TRTABFU (option)

. Allocation of backup file:
          $NMBU.$TRTABTC
```

WARNING:

```
If you want to use tapes for this file, you must enter
the tape names in the parameters of the PTINALSV JCL.

. Allocation of program libraries:
          $NMLI.$LIBLMT
          $NMLI.$LIBCUBT
          $NMLI.$LIBCUTT
          $NMLI.$LIBCUPT
          $NMLI.$LIBCU20
          $NMLI.$LIBINVT
          $NMLI.$LIBSUT (batch transaction library)
```

To run Pactables under a TDS different from that of VA Pac, the parameter values must be such that library names differ from VA Pac TDS library names.

2. TDS PREPARATION:

This preparation is to be performed only if Pactables is not integrated into an alrealdy existing TDS.

The TDS preparation is performed by the execution of the 'MTPREP' system procedure provided by the vendor. The 'PTINMPRE' member located in the '$NMLI.$LIBJCLT' library includes the execution of the 'MTPREP' procedure supplied by the vendor in the SYS.HSLLIB library.

3. INSTALLATION OF FILES AND PROGRAMS:
(see the JCL at the end of this sub-chapter)

Files and programs are installed by the execution of the 'PTINUNLD' member located in the $NMLI.$LIBJCLT library. This member is actually a sequence of LIBMAINT and CREATE, and can logically be partitioned in the following way:

```
. Unloading of the TDS and first TPR sources:
          $NMTD.$LIBSLT

. Program unloading (compile-units):
          $NMLI.$LIBCUBT
          $NMLI.$LIBCUTT

. Copy of sub-program CUs in the permanent library:
          $NMLI.$LIBCUPT

. Copy of specific V2.0 functions' CUs:
          $NMLI.$LIBCU20

. Unloading of UFAS table files:
          $NMTU.$TRTABTE
          $NMTU.$TRTABFU (option)

. Unloading of table backup file:
          $NMBU.$TRTABTC
```

4. GENERATION OF PACTABLES TDS:

This paragraph only describes the generation of a Pactables TDS which is independent or inserted in a VA Pac TDS.

The generation is executed by the PTINMGEN procedure included in the $NMLI.$LIBJCLT library.

The TDS source supplied in $NMTD.$LIBSLT corresponds to a TDS which contains the Pactables function only.

Note:

The member which contains the TDS source is:
STDST for the Pactables function.
You must copy or rename this source under the 'STDS' reserved name before executing the procedure.

If you want to integrate VA Pac and Pactables in the same TDS, a corresponding TDS source, named STDSPT, is supplied in the VA Pac system SL library. In this case, you must copy or rename this source under the 'STDS' reserved name before executing the procedure.
You may have to change the 'PROGRAM-ID' clause by the value given to $NMTD (TDST by default).

5. PROGRAM LINK-EDIT

(see the JCL at the end of this subchapter)

To link-edit programs, you must execute the PTINBLNK member for batch procedures and the PTINTLNK member for on-line programs.

6. RESTORATION OF A PACTABLES 2.0 TEST DATABASE

To create a Pactables test database, you must execute the PTINRST1 member of the $NMLI.$LIBJCL library (Pactables restoration procedure). Its input is the backup coming from the installation tape.

7. USER PARAMETER UPDATE

The system can run only if the user parameters have been given to Pactables. Before testing, you must update user parameters in TG via the PMTA procedure. (an initial general user code is supplied, at installation time, in the TG file: '********SUPER').

8. TDS SUBMISSION

To submit the TDS, run the PTEXTDPT procedure. The '&1' parameter must be set to 'STEP1' to load the TPRs in BACKING STORE.

9. TUF

2 sub-programs, TUF000 and TUF900 must be declared in TDS (addition of 2 USE cards to the source) when executing applications using the TUF function. The TB work file can be reinitialized by the PTEXINTB procedure between two utilizations.

20.5. INSTALLATION JCL PTINPRPT

```
COMM 'VA-PACTABLE 2.5     ';
COMM '************************************';
COMM '*    FILE AND LIBRARY              *';
COMM '*    ALLOCATION                    *';
COMM '*                                  *';
COMM '************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
COMM '*** TDS    ***';
IV   PTINALTD,($NMLI.$LIBJCLT,&RFLI);
IV   PTINALTE,($NMLI.$LIBJCLT,&RFLI);
IV   PTINALTG,($NMLI.$LIBJCLT,&RFLI);
IV   PTINALTV,($NMLI.$LIBJCLT,&RFLI);
COMM '*** BATCH  ***';
IV   PTINALSV,($NMLI.$LIBJCLT,&RFLI);
COMM '*** BIBLIOTHEQUES ***';
IV PTINALLI,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBLMT,LM,5,1,130);
IV PTINALLI,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBCUBT,CU,5,1,150);
IV PTINALLI,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBCUTT,CU,5,1,150);
IV PTINALLI,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBCUPT,CU,1,1,10);
IV PTINALLI,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBCU20,CU,1,1,10);
IV PTINALLI,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBSUT,SL,1,1,10);
IV PTINALIV,($NMLI.$LIBJCLT,&RFLI)
   VL=($NMLI.$LIBINVT,SL,1,1,20);
```

20.6. INSTALLATION JCL PTINMPRE

```
COMM 'VA-PACTABLE 2.5     ';
COMM '************************************************';
COMM '*                                              *';
COMM '*   PREPARATION PROCEDURE FOR TDS              *';
COMM '*                                              *';
COMM '************************************************';
OVL  HOLD;
VL   PRY='SYSFILE=CAT,FILESTAT=CAT,CATNAME=$CTNM,IMPORT=NO',
     PRN='SYSFILE=RSD,FILESTAT=UNCAT',
     FF='$NMTD,$DVTD,$MDTD,$DVTD,$MDTD,DEAL=Y',
     GG='DBGSZ=1,MAXDBG=3,CBLSZ=1,SMSZ=15,MAXSM=20',
     VLVL='VL=('&FF','&PR$CTTD'';
IV   TP7PREP SYS.HSLLIB &VLVL,&GG);
SEND '===>  PREPARATION OF '$NMTD' SUCCESSFUL  <===';
```

20.7. INSTALLATION JCL PTINUNLD

```
COMM 'VA-PACTABLE 2.5     ';
COMM '************************************';
COMM '*    INSTALLATION TAPE UNLOADING    *';
COMM '*            LIBRARIES              *';
COMM '*           SYSTEM FILES            *';
COMM '************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    DVT='DVC=$DVTP',MDT='MD=$TAPE';
OVL HOLD;
LMN  SL,IF=(SV.SLT,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=($NMTD.$LIBSLT,&RFLI),
     COM='MV INFILE:*,REPLACE;';
LMN  CU,IF=(SV.CUBT,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=($NMLI.$LIBCUBT,&RFLI),
     COM='MV INFILE:*,REPLACE;';
LMN  CU,IF=(SV.CUTT,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=($NMLI.$LIBCUTT,&RFLI),
     COM='MV INFILE:*,REPLACE;';
LIB  CU,IL1=($NMLI.$LIBCUTT,&RFLI),
        IL2=($NMLI.$LIBCUBT,&RFLI);
LMN  CU,LIB=($NMLI.$LIBCUPT,&RFLI),
     COM='MV IL1:ZAR980;STATUS RESET;
          MV IL1:ZAR985;STATUS RESET;
          MV IL1:ZTACCE;STATUS RESET;
          MV IL1:PAP830;STATUS RESET;
          MV IL1:PAP930;STATUS RESET;
          MV IL2:PTA800;STATUS RESET;
          MV IL2:PTA900;STATUS RESET;';
CR   IF=(SV.TE,&DVT,&MDT,FSN=ANY,END=LEAVE),
     OF=($NMTU.$TRTABTE,&RFTU);
CR   IF=(SV$LANG.TC,&DVT,&MDT,FSN=ANY,END=LEAVE),
     OF=($NMBU.$TRTABTC,&RFBU);
LMN  CU,IF=(SV.CUT20,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=($NMLI.$LIBCU20,&RFLI),
     COM='MV INFILE:*,REPLACE;';
```

20.8. INSTALLATION JCL PTINMGEN

```
COMM 'VA-PACTABLE 2.5     ';
COMM '************************************************';
COMM '*                                              *';
COMM '*   TDS SYSTEM GENERATION                      *';
COMM '*                                              *';
COMM '************************************************';
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     TDDVN=$DVTD,TDMDN=$MDTD,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     LMDVN=$DVLI,LMMDN=$MDLI,
     CTTDN='FILESTAT=UNCAT,DVC=$DVTD,MD=$MDTD',
     RFTD=&CTTD$CTTD,
     RFTM='DVC=$DVTM,MD=$MDTM';
OVL  HOLD;
LIB  CU,INLIB1=($NMLI.$LIBCUPT,&RFLI);
LMN  CU,LIB=TEMP,
     COM='MV IL1:ZAR980;STATUS RESET;
          MV IL1:ZAR985;STATUS RESET;
          MV IL1:ZTACCE;STATUS RESET;
          MV IL1:PAP830;STATUS RESET;
          MV IL1:PAP930;STATUS RESET;';
IV   TP7GEN SYS.HSLLIB
     VL=($NMTD,&TDDV$CTTD,&TDMD$CTTD,,,,,,$NMLI.$LIBLMT,
        LMDVC=&LMDV$CTLI,LMMD=&LMMD$CTLI);
END:
```

20.9. INSTALLATION JCL PTINTLNK

```
COMM 'VA-PACTABLE 2.5     ';
COMM '*****************************************';
COMM '*                                       *';
COMM '*   LINKING OF ALL ON-LINE PROGRAMS     *';
COMM '*                                       *';
COMM '* $TPR0: PACTABLE TPR NAME IN SM LIBRARY *';
COMM '*                                       *';
COMM '*   PLEASE KEEP ALL LINKING REPORTS     *';
COMM '*   ============================        *';
COMM '*****************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM',
    TPR='$TPR0';
OVL HOLD;
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTPAA0,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTPLNK,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP500,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP510,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP520,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP530,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP540,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP550,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP560,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP570,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP580,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP590,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP599,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP600,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP610,&TPR);
IV   PTINTLSO,($NMLI.$LIBJCLT,&RFLI),VL=(ZTP620,&TPR);
END:
```

20.10. INSTALLATION JCL PTINBLNK

```
COMM 'VA-PACTABLE 2.5     ';
COMM '*******************************************';
COMM '*                                         *';
COMM '*   LINKING OF ALL BATCH PROGRAMS         *';
COMM '*                                         *';
COMM '*   PLEASE KEEP ALL LINK REPORTS          *';
COMM '*   ============================          *';
COMM '*******************************************';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAD05;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAD10;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAD20;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAINI;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTARSD;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTARSG;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTARSV;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTARTG;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTASVD;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTASVG;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTASVV;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAU80;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAV10;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAV20;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA100;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA120;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA150;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA160;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA250;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA290;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA300;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA310;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA320;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA350;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA360;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA400;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA410;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA420;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTA430;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAR20;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAXVD;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAXVG;
IV   PTINBLSO,($NMLI.$LIBJCLT,&RFLI),VL=PTAXVV;
```

20.11. INSTALLATION JCL PTININTJ

```
COMM 'VA-PACTABLE 2.5     ';
0000001PTEXJOBL:$NMLI.$LIBJCLT:$MDLI:$DVLI
0000001PTEXJOBL:$NMLI.$LIBJCLT
```

20.12. INSTALLATION JCL PTEXTDPT

```
COMM 'VA-PACTABLE 2.5     ';
COMM '*****************************************';
COMM '*                                       *';
COMM '* EXECUTION OF TDS : $NMTD   PACTABLE     *';
COMM '* &1 = BACKING-STORE : STEP1 , ELSE STEP2 *';
COMM '*                                       *';
COMM '*****************************************';
MVL STEP1,START='WARM',
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',RFTU=&CTTU$CTTU,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',RFBU=&CTBU$CTBU,
    CTTDN='FILESTAT=UNCAT,DVC=$DVTD,MD=$MDTD',RFTD=&CTTD$CTTD,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
OVL HOLD;
JUMP &1;
STEP1:
LIB  SM,INLIB1=($NMTD.$LIBSMT,&RFTD);
SYSMAINT COMFILE=*DEMER;
$IN DEMER;
SM;
LOAD MODULE=$TPR0 INPUT=INLIB1 REPLACE;
$EIN DEMER;
STEP2:
IV   PTINALTJ ($NMLI.$LIBJCLT,&RFLI);
IV   PTINALTB ($NMLI.$LIBJCLT,&RFLI);
JOBLIB SM,$NMTD.$LIBSMT;
STEP $NMTD,FILE=($NMLI.$LIBLMT,&RFLI),DUMP=DATA,OPTIONS=&START;
     SZ  150,POOLSZ=70,NBBUF=70;
     ASG DBUGFILE,$NMTD.DEBUG,&RFLI,
         SHARE=DIR;
     ASG PT80TD,$NMTU.$TRTABTD,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PT80TD,NBBUF=2,JOURNAL=BEFORE;
     ASG PT80TE,$NMTU.$TRTABTE,&RFTU,
         SHARE=MONITOR;
     DEF PT80TE,NBBUF=2,JOURNAL=BEFORE;
     ASG PT80TJ,$NMTU.$TRTABTJ,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PT80TJ,NBBUF=2,JOURNAL=BEFORE;
     ASG PT80TG,$NMTU.$TRTABTG,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PT80TG,NBBUF=2,JOURNAL=BEFORE;
     ASG PT80TV,$NMTU.$TRTABTV,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PT80TV,NBBUF=2,JOURNAL=BEFORE;
     ASG PT80TB,$NMTU.$TRTABTB,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PT80TB,NBBUF=2,JOURNAL=BEFORE;
ESTP;
```

20.13. LIBRARY ALLOCATION JCL

```
COMM 'VA-PACTABLE 2.5     ';
COMM  '*****************************************';
COMM  '*                                       *';
COMM  '*      PACBASE  LIBRARY  ALLOCATION     *';
COMM  '*                                       *';
COMM  '*****************************************';
MVL    CTLI='FILESTAT=CAT,UNCATNOW',
       CLIY='FILESTAT=CAT,DVC=$DVLI,MD=$MDLI',
       DV='DVC=$DVLI',MD='MD=$MDLI';
OVL    HOLD;
JUMP   CAT$CTLI;
CATY:
DALC   &1,&CTLI;
JUMP   CONTINUE;
CAT    &1,TYPE=FILE,SHARE=UNSPEC;
LALC   &2,(&1,&CLIY,SZ=(&3,&4)),MEMBERS=&5;
JUMP   END;
CATN:
DALC   &1,&DV,&MD;
JUMP   CONTINUE;
LALC   &2,(&1,&DV,&MD,SZ=(&3,&4)),MEMBERS=&5;
END:
```

```
   COMM 'VA-PACTABLE 2.5     ';
   MVL    CTLI='FILESTAT=CAT,UNCATNOW',
          CLIY='FILESTAT=CAT,DVC=$DVLI,MD=$MDLI',
          DV='DVC=$DVLI',MD='MD=$MDLI';
   OVL    HOLD;
   JUMP   CAT$CTLI;
   CATY:
   CAT    &1,TYPE=FILE,SHARE=UNSPEC;
   JUMP   CONTINUE;
   LALC   &2,(&1,&CLIY,SZ=(&3,&4)),MEMBERS=&5;
   JUMP   CONTINUE;
   LIB    SL,INLIB1=($NMLI.$LIBJCLT,&RFLI);
   LMN    SL,LIB=&1,
          COM='MV IL1:PTIV*,REPLACE;';
   JUMP   END;
   CATN:
   LALC   &2,(&1,&DV,&MD,SZ=(&3,&4)),MEMBERS=&5;
   JUMP   CONTINUE;
   LIB    SL,INLIB1=($NMLI.$LIBJCLT,&RFLI);
   LMN    SL,LIB=(&1,&DV,&MD),
          COM='MV IL1:PTIV*,REPLACE;';
   END:
```

20.14. FILE ALLOCATION JCL

```
COMM 'VA-PACTABLE 2.5     ';
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     CTTUY='FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
     CTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     CTUY='FILESTAT=CAT',RTU=&CTU$CTTU,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     DV1='DVC=$DVTU',MD1='MD=$MDTU',
     CATFN='FILESTAT=UNCAT',
     CATFY='FILESTAT=CAT';
DALC $NMTU.$TRTABTD,&RFTU;
JUMP CONTINUE;
JUMP TRTABTD$CTTU;
TRTABTDY:
CAT  $NMTU.$TRTABTD,TYPE=FILE,SHARE=UNSPEC;
TRTABTDN:
PALC $NMTU.$TRTABTD,
     UNIT=CYL,&DV1,GBL=(&MD1,SZ=2),INCRSZ=1,
     UFAS=(INDEXED=(CISZ=4096,RECSZ=240,KEYLOC=1,KEYSZ=21,
     CIFSP=20,CAFSP=20)),
     &CATF$CTTU;
```

```
    COMM 'VA-PACTABLE 2.5     ';
    MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTTUY='FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
         CTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTUY='FILESTAT=CAT',RTU=&CTU$CTTU,
         CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
         RFLI=&CTLI$CTLI,
         DV1='DVC=$DVTU',MD1='MD=$MDTU',
         CATFN='FILESTAT=UNCAT',
         CATFY='FILESTAT=CAT';
    DALC $NMTU.$TRTABTE,&RFTU;
    JUMP CONTINUE;
    JUMP TRTABTE$CTTU;
    TRTABTEY:
    CAT  $NMTU.$TRTABTE,TYPE=FILE,SHARE=UNSPEC;
    TRTABTEN:
    PALC $NMTU.$TRTABTE,
         UNIT=CYL,&DV1,GBL=(&MD1,SZ=2),INCRSZ=1,
         UFAS=(INDEXED=(CISZ=2048,RECSZ=90,KEYLOC=1,KEYSZ=17)),
         &CATF$CTTU;
```

```
    COMM 'VA-PACTABLE 2.5     ';
    MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTTUY='FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
         CTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTUY='FILESTAT=CAT',RTU=&CTU$CTTU,
         CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
         RFLI=&CTLI$CTLI,
         DV1='DVC=$DVTU',MD1='MD=$MDTU',
         CATFN='FILESTAT=UNCAT',
         CATFY='FILESTAT=CAT';
    DALC $NMTU.$TRTABTG,&RFTU;
    JUMP CONTINUE;
    JUMP TRTABTG$CTTU;
    TRTABTGY:
    CAT  $NMTU.$TRTABTG,TYPE=FILE,SHARE=UNSPEC;
    TRTABTGN:
    PALC $NMTU.$TRTABTG,
         UNIT=CYL,&DV1,GBL=(&MD1,SZ=2),INCRSZ=1,
         UFAS=(INDEXED=(CISZ=2048,RECSZ=85,KEYLOC=1,KEYSZ=22,
         CIFSP=20,CAFSP=20)),
         &CATF$CTTU;
```

```
    COMM 'VA-PACTABLE 2.5     ';
    MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTTUY='FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
         CTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTUY='FILESTAT=CAT',RTU=&CTU$CTTU,
         CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
         RFLI=&CTLI$CTLI,
         DV1='DVC=$DVTU',MD1='MD=$MDTU',
         CATFN='FILESTAT=UNCAT',
         CATFY='FILESTAT=CAT';
    DALC $NMTU.$TRTABTV,&RFTU;
    JUMP CONTINUE;
    JUMP TRTABTV$CTTU;
    TRTABTVY:
    CAT  $NMTU.$TRTABTV,TYPE=FILE,SHARE=UNSPEC;
    TRTABTVN:
    PALC $NMTU.$TRTABTV,
         UNIT=CYL,&DV1,GBL=(&MD1,SZ=2),INCRSZ=1,
         UFAS=(INDEXED=(CISZ=4096,RECSZ=1059,KEYLOC=5,KEYSZ=35,
         RECFORM=V,CIFSP=20,CAFSP=20)),
         &CATF$CTTU;
```

```
    COMM 'VA-PACTABLE 2.5     ';
    MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTTUY='FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
         CTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
         CTUY='FILESTAT=CAT',RTU=&CTU$CTTU,
         CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
         RFLI=&CTLI$CTLI,
         CTSTN=3,CTSTY=4,
         RFST=&CTST$CTLI,
         DV1='DVC=$DVTU',MD1='MD=$MDTU',
         CATFN='FILESTAT=UNCAT',
         CATFY='FILESTAT=CAT';
    DALC $NMTU.$TRTABTJ,&RFTU;
    JUMP CONTINUE;
    JUMP TJ$CTTU;
    TJY:
    CAT  $NMTU.$TRTABTJ,TYPE=FILE,SHARE=UNSPEC;
    TJN:
    PALC $NMTU.$TRTABTJ,
         UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
         UFAS=(RELATIVE=(CISZ=2048,RECSZ=80)),
         &CATF$CTTU;
    CR   IF=($NMLI.$LIBJCLT,&RFLI,SUBFILE=PTININTJ),
         OF=($NMTU.$TRTABTJ,&RTU),START=&RFST,HALT=1,
         COMFILE=($NMLI.$LIBJCLT,&RFLI,SUBFILE=PTEXPDSL);
```

```
   COMM 'VA-PACTABLE 2.5     ';
   MVL  CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
        CTBUY='FILESTAT=CAT,UNCATNOW',RFBU=&CTBU$CTBU,
        CBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
        CBUY='FILESTAT=CAT',RBU=&CBU$CTBU,
        CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
        RFLI=&CTLI$CTLI,
        DV1='DVC=$DVBU',MD1='MD=$MDBU',
        DVT1='DVC=MT/T9',MDT1='MD=TAPE1',
        DVT2='DVC=MT/T9',MDT2='MD=TAPE2',
        CATFN='FILESTAT=UNCAT',
        CATFY='FILESTAT=CAT';
   DALC $NMBU.$TRTABTC,&RFBU;
   JUMP CONTINUE;
   DALC $NMBU.$TRTABTC/G-1,&RFBU;
   JUMP CONTINUE;
   UNCAT $NMBU.$TRTABTC,TYPE=FILE;
   JUMP CONTINUE;
   JUMP TC$CTBU;
   TCY:
   CAT  $NMBU.$TRTABTC,TYPE=FILE,NBGEN=2,SHARE=UNSPEC;
   JUMP TC$MDSV;
   TCD:
   PALC $NMBU.$TRTABTC/G+1,
        UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
        UFAS=(SEQ=(CISZ=$CISEQ,RECSZ=1063,RECFORM=V)),
        &CATF$CTBU;
   PALC $NMBU.$TRTABTC/G+1,
        UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
        UFAS=(SEQ=(CISZ=$CISEQ,RECSZ=1063,RECFORM=V)),
        &CATF$CTBU;
   JUMP ENDTC;
   TCT:
   PALC $NMBU.$TRTABTC/G+1,
        &DVT1,GBL=(&MDT1,SZ=1),
        UFAS=(SEQ=(BLKSZ=10630,RECSZ=1063,RECFORM=VB,NBSN)),
        &CATF$CTBU;
   PALC $NMBU.$TRTABTC/G+1,
        &DVT2,GBL=(&MDT2,SZ=1),
        UFAS=(SEQ=(BLKSZ=10630,RECSZ=1063,RECFORM=VB,NBSN)),
        &CATF$CTBU;
   JUMP ENDTC;
   TCN:
   JUMP TCN$MDSV;
   TCND:
   PALC $NMBU.$TRTABTC,
        UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
        UFAS=(SEQ=(CISZ=$CISEQ,RECSZ=1063,RECFORM=V)),
        &CATF$CTBU;
   TCNT:
   ENDTC:
```

20.15. INSTALLATION OF THE PRINT JCL

INSTALLATION OF THE TABLE PRINT JCL

This installation complements the installation to enable the submission of the table print procedure (PRTA) in on-line mode.

```
    The print request control cards are inserted:
   - either in batch mode by the PMTA proc. ('TJ' line code)
   - or in on-line mode via the 'LJ' screen (see example on
     next page)

Line numbers enable to classify the inserted JCL lines:
     < 600000  JCL line at the beginning of the flow
     > 599999  JCL lines at the end of the flow
```

PACBASE-TABLES 8.0 JCL OF PRINT REQUESTS USER :

```
A CO LINE         CONTENT
  TJ 000100       MVL CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
  TJ 000200           RFLI=&CTLI$CTLI,LIST='LIST=ALL';
  TJ 000300       IV  PTUSPRTA,($NMLI.$LIBJCLT,&RFLI),$LIST,
  TJ 000400           VL=(SIZEED=10,SIZEET=10);
  TJ 000500       $IN PRTA;
  TJ 600100       $EIN PRTA;
```

O : LJ KEY : JOB :

21. UTILIZATION TESTS

USE TESTS

These test jobs include the following steps:

. On-line use tests under TDS,
. Test on batch updating, printing and reorganization,
. Test on table generation.

The test set comprises three tables:

. 'TEMPER' without historical account,
. 'CUSTOM' with two historical accounts 01/03/1985 and 15/01/1985,
. 'ARTICL' with historical account 15/01/1987.

On-line tests:

Open Pactables test files under TDS.

Consult all the screens.

Execute some updates.

Batch tests:

Execute the PRTA procedure.

Execute the EXTA procedure.

Close the Pactables files under TDS.

Execute the UPTA procedure.

Reorganization of test table:

.Save TV and TD files (CREATE)

.Execute the reorganization (RETA) which comprizes:
.Reorganization of TV file (Prog. PTA400 and PTA410),
.Reorganization of TD file (Prog. PTA420),
.Constitution of the TC backup file (Prog. PTA430),

.Execute a printing (PRTA) for validation.

.Open the files under TDS and perform some validation tests after reorganization in on-line mode.

Table generation tests (GETT procedure) :

Close the files under TDS.

Execute an extraction under VA Pac (GETA or GETD).

Execute the GETT procedure.

Verify the execution.

Open the files again under TDS and perform some valida- tion tests.

22. PACTABLES STANDARD REINSTALLATION

STANDARD REINSTALLATION

The Pactables function should be reinstalled whenever a new sub-release is shipped containing improvements upon the preceding release.

The new version is identified by a number and is delivered in the following form:

. A complete installation tape of the product,
. The list of corrected bugs,
. Possibly, a note completing the set of instructions described in this subchapter.

Generally, only the program libraries and the system files are affected by the new version.

Therefore, re-installation most often involves execution of the procedures used for the first installation.

It uses a complete installation tape whose content is described in the previous subchapter.

Reinstallation includes the following steps:

```
. Installation tape backup,
. Reallocation of system files,
. System file reallocation,
. Tape unloading,
. TDS regeneration,
. Program link-edit,
. Taking into account the new Pactables error message
  file.
```

REALLOCATION OF SYSTEM FILES

To re-install a version, you must always retrieve the new system files. Since their size varies according to the versions, you must re-allocate them by running the PTINPRPT procedure.

INSTALLATION TAPE UNLOADING:

The unloading, executed by the PTINUNLD procedure, is described in chapter 'INSTALLATION'.

WARNING: THIS PROCEDURE INITIALIZES THE TD AND TV FILES.

TDS GENERATION:

The TDS must be generated to take into account the new version of the Pactables routines used in on-line transactions (ZAR980, ...).

PROGRAM LINK-EDIT:

Execution of the following link-edit procedures:

```
   . PBINTLNK: TPRs link-edit,
```

. PBINBLNK: batch program link-edit.