VisualAge Pacbase 2.5

PACTABLES 2.5 – IBM VSE, CICS
OPERATIONS MANUAL

DETDO000251A

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 (March 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

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.

This manual contains four sections:

1. General presentation of the Pactables function

   Chapter 1 - Pactables Components

   Chapter 2 - Environment

2. Batch operation procedures

   Chapters 4-18: Batch Procedures

3. Retrieval from previous releases of the Pactables function

   Chapter 19 - 7.x Table Retrieval

   Chapter 20 - 8.x Table Retrieval

4. Installation

   Chapter 21 - Installation

For a more detailed description of each chapter, refer to the Table of Contents.

The index at the end of the manual will help you find information on installation and running of the system.

HOW TO USE THIS MANUAL FOR SYSTEM INSTALLATION

After reading the introductory chapters to the system: INTRODUCTION, COMPONENTS, and ENVIRONMENT, go directly to the INSTALLATION chapter.

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)

LIBRARIES

The Pactables function operates in both on-line and batch mode, using the following libraries:

- A library for on-line programs,

- A library for batch programs,

- A library for SLI members,

- A library for VSE procedures.

NOTE:

When Pactables is operated on the same site as VA Pac, its programs and parameters may be stored in the VA Pac System libraries.

In this case, the size of each VA Pac System library must be increased by the size of the corresponding Pactables library specified below.

2.2. THE ON-LINE PROGRAM LIBRARY

THE ON-LINE PROGRAM LIBRARY

Its size is approximately 1200 K or 24 tracks of a 3380 disk unit. It contains the following programs :

CODE	OPERATION AND MEANING
FT00	Tables extraction (TUF-TP MODULE)
FT90	User Interface (TUF-TP MODULE)
P500	Enter, FT or 'clear' : initial screen
P510	C1: read-write access to mono-item with CR, MO, DE for updating
P512	C1: read-write access to mono-item with CR, MO, DE for updating (V.2)
P520	C2: read-only access to multi-item with DE for an item deletion
P522	C2: read-only access to multi-item with DE for an item deletion (V.2)
P530	LT : list of tables
P540	LS : list of sub-schemas/sub-systems
P550	LD : list of documentation
P560	C3 : read-only access to item hist. acc.
P570	HELP screen
P580	LH : list of historical accounts
P590	LJ, LE : print requests
P599	Display system errors
P600	Parameter & password updating
P610	User code updating
P620	Access authorization updating
P820	Optimized generalized access module
P920	User generalized access module
PLNK	Pactables access via user-program module

NOTE:

Codes of programs and maps are given without the 2-character suffix which is made up of the two first characters of the transaction code used to access the Pactables function.

IMPORTANT

Two additional programs (P512 and P522) have been supplied since Pactables Release 2.0.

During updates, the P510 and P520 programs may call the user check routines in order to perform additional checks. As a default, the generation option of these routines is without the century management.

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

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

2.3. THE BATCH PROGRAM LIBRARY

THE BATCH PROGRAM LIBRARY

Size: approximately 2700 K or 57 tracks of a 3380 disk unit.

CODE	PROC.	OPT.	MEANING
PTU001			Transaction file copy
PTAINI	INTA		File initialization
PTARSD	RSTA		Table restoration
PTARSV	-		- -
PTARSG	-		- -
PTARTG	R2TA		Retrieval 7.2 tables
PTAR20	RTTA		Retrieval 8.02, 1.2 tables ->2.5
-	R2TA		
-	R3TA		Retrieval 7.3 tables
PTASVD	SVTA		Table backup
PTASVV	-		- -
PTASVG	-		- -
-	RETA		Table reorganization
PTAU80	TUTA		Direct consultation of tables
PTAXVD	R2TA		- - -
-	R3TA		- - -
PTAXVV	R2TA		- - -
-	R3TA		- - -
PTAXVG	R2TA		- - -
-	R3TA		- - -
PTA100	PMTA		Parameter update
PTA120	-		- -
PTA150	EXTA		Table extraction
PTA160	-		- -
PTA250	GETT		Table generation
PTA290	-		- /Lists -
-	LDTA		- /Lists -
PTA300	UPTA		Table update
PTA302	-		- -
PTA310	IMTA		Table import
PTA312	-		- -
PTA320	PRTA		Table printout
PTA350	UPTA		- -
-	IMTA		- -
-	PRTA		- -
PTA360	UPTA		- -
-	IMTA		- -
-	PRTA		- -

CODE	PROC.	OPT.	MEANING
PTA400	RETA		Table reorganization
PTA410	-		- -
PTA420	-		- -
PTA430	-		- -
PTAD05	CDT1	DTM	Table-description comparison
PTAD10	-	-	- - -
PTAD20	CDT2	-	- - -
PTAV10	CVTA	-	Table update
PTAV20	-	-	- -
PTA800			Optimized access module
PTA900			Generalized access module
PTATCD	TCTA		TD file sort
PTATCG	-		TG file sort
PTATCV	-		TV file sort
PTATC1	-		TC partitioning according to the type of file
PTATC2	-		Sorted TC file rebuilding

<u>IMPORTANT NOTE:</u>

Two other programs (PTA302 and PTA312) have been supplied since version 2.0.

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

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

THE LIBRARY OF COBOL/VS SUB-PROGRAMS

It contains the following programs:

CODE	MEANING
....	.... Batch sub-programs ....
PTA800	Optimized access module
PTA900	Generalized access module
....	.... On-line sub-programs ....
P820	Optimized access module
P920	User generalized access module

These sub-programs must be used with programs developed in Cobol/VS.

2.4. THE VA Pac MACROS-STRUCTURES LIBRARY

VA PAC MACRO-STRUCTURES LIBRARY

The Macro-structures are the following ones:

CODE	MEANING
AATUFA	Description of the table data element
AATUFL	'LT' or 'LH' list
AATUFS	'LS' or 'LC' list
AATUFX	List of items

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 xxFT90 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.

2.5. THE PARAMETER LIBRARY

THE SOURCE LIBRARY OF SLI MEMBERS

Its size does not exceed one track of a 3380 disk unit.

It contains the JCL members or IDCAMS commands inserted into$ SLI').

- DFxxyyff contains the IDCAMS commands for defining the database files,

```
         where: ff   = the file code,
                xxyy = a prefix made up of the system root and of
                       a code representing either a system or
      user
                       file.
```

- RGxxyyff contains the IDCAMS commands for rotating the back-up files, permitting the centralized management of the number of generations saved.

- DBJxxyy contains the JCL inserted at the beginning of each job (resource definitions, standards).

- ppppxxyy contains the JCL for certain procedures, where: pppp = the procedure code.

In order to simplify installation, it is preferable to use an existing library already declared in the LIBDEF SOURCE,SEARCH of the POWER JCL.

2.6. PROCEDURE LIBRARY

THE (VSE) PROCEDURE LIBRARY

Its size does not exceed one track of a 3380 disk unit.

It contains the procedures for parameterized JCL and for defining sort space or non-VSAM work space.

It must be shared by VA Pac, Pactables, and DSMS in order to avoid work space conflicts between partitions.

2.7. SYSTEM FILES

THE 'SYSTEM' FILES

These constitute the actual system. They are not affected by daily manipulations and must be reloaded whenever the system is reinstalled.

These are:

```
 . The libraries mentioned in the previous subchapters,

 . A file (TE) containing error messages and documentation:

            .Size         : Approximately 1000 records
            .Organization : VSAM-KSDS
            .Recsize      : 90
            .Key          : 17 (position 0)
            .Utilization  : batch and on-line
```

2.8. EVOLVING FILES

EVOLVING FILES

These files contain user information and are managed by the system in batch and on-line modes.

The first two files make up the actual Pactables database.

.TABLE DESCRIPTION FILE (TD)

```
     .Organization : VSAM-KSDS
     .Recsize      : 240
     .CI size      : 1024
     .Key          : 19 (position 0)
     .Utilization  : Batch and on-line
     .Space        : 4 records per C.I. of 1024.
```

.TABLE CONTENTS FILE (TV)

```
     .Organization : VSAM-KSDS
     .Recsize      : 80 to 1100
     .CI size      : 1024
     .Key          : 33 (position 4)
     .Utilization  : Batch and on-line
     .Space        : Varies with table's length.
```

.USER PARAMETER FILE (TG)

It contains:
- the user codes and related access authorizations,
- user JCL for table printing.

```
     .Organization : VSAM-KSDS
     .Recsize      : 85
     .CI size      : 1024
     .Key          : 22 (position 0)
     .Utilization  : Batch and on-line
     .Space        : 12 records per C.I. of 1024.
```

.BACKUP FILE (TC)

This file contains the backup of Pactables user files.

.Organization	: SAM-ESDS or SAM
.Recsize	: (572 32760)
.CI size	: 8192
.Utilization	: Batch

.WORK FILE (TB)

This file, used by the TUF-TP function, makes it possible to temporarily store the data extracted from Pactables, the data used for Pactables update, and the update error messages.

.Recsize	: 80 to 1140
.Organisation	: VSAM-KSDS
.Recsize	: 80 to 1140
.CI size	: 1024
.key	: 63 (position 3)
.Utilization	: on-line
.Space	: Varies with table's lenght.

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 CICS/VSE (Command level) Rel 1.7 or higher.

It must include BMS (the PAGING option is not used) with the option MAPALGN=NO.

It must include the option SPOOL=YES to submit batch jobs (LJ screen).

The table description (TD), the table contents (TV) and the table user parameters (TG) files are protected by the 'DYNA- MIC TRANSACTION BACKOUT' (ROLLBACK) option.

The use of the CICS 'EMERGENCY RESTART' option is recommen- ded.

The size of the COMMAREA in use is 8 K.

GENERAL INFORMATION - HOW THE SYSTEM RUNS

The general characteristics of the system are as follows:

.A transaction code is used to access the system. Branching to the different programs is ensured by 'RETURN TRANSID'.

Each screen is associated with a transaction code and a program (for example: XXP510 performs updates, and the transaction code is XX01 for a mono-item screen).

.In case of a system ABEND, an 'ABEND MAP' is generated. The display program for this map (XXP599) is called by XCTL.

PROCESSING UPPER AND LOWER CASE

The Table function automatically turns lowercase into upper- case in screens with the UCTRAN option in the TCT of fields where uppercase is required, i.e. the user code and its as- sociated password and the operation code.

3.3. ACCESS METHODS

ACCESS METHODS (VSAM)

The Pactables function manages its files using the VSAM-KSDS indexed access method without secondary index.

Files are protected against concurrent read-write accesses (SHARE OPTION 2). In particular, the Pactables function protects itself against simultaneous updates in both batch and on-line modes by using this share option.

Each batch procedure includes a DELETE/DEFINE step in case of file reloading. As a result, files need not include the REUSE default option and, consequently, may be allocated using either the UNIQUE option or the SUB-ALLOCATION option.

3.4. BATCH ENVIRONMENT

TEMPORARY FILES

In batch mode, the system runs using both the standard functions of the operating system and the VSAM access method.

The internal work files of procedures are mostly of the SAM-ESDS type. They are defined and deleted for each utilization.

PARTITION SIZE

The memory size necessary to the execution of the batch procedures varies mostly according to the size of the bufers allocated to the files used by those procedures.

Three parameters are of particular significance:

- The size of the EXEC Card, given in the parameter SIZE, which relates to the loading area of the executable load- modules.

- The size of the GETVIS AREA, used especially by VSAM for buffers.

- The total size of the partition, defined for the system, and which is the sum of the two preceding areas.

TAPE USAGE

Files that are likely to be large can be re-directed onto tapes.

In procedures, the SYSnnn logical unit associated to the file(s) --which must be assigned to a tape drive-- is specified by a comment.

For files re-directed onto a tape, the maximum block size is used. Therefore, files with smaller block sizes are automatically accepted.

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)

.Pactables database migration (TCTA)

.Table restoration (RSTA)

.Printing of table description lists (LDTA)

.Update of user parameters (PMTA)

.Extraction of data (EXTA)

.Direct reading of tables (TUTA)

With the Dispatched Table Management option (DTM):

.Table description comparison (CDT1, CDT2)

.Table extraction for update (CVTA).

For retrieval of previous releases:

.Retrieval of 7.2 Pactables files (R2TA)

.Retrieval of 7.3 Pactables files (R3TA)

.Retrieval of the 8.xx or 1.2 backup file (RTTA).

NOTE

Pactables does not provide a journal of update transactions.

4.3. ABNORMAL EXECUTIONS

ABNORMAL EXECUTIONS

A batch program execution may terminate abnormally. For example, input-output errors on the system files or on the database cause a forced abnormal end with a USER ABEND, accompanied by a message on the SYSLST file.

When an ABEND occurs, the user must first find this message. It is displayed in the following manner:

```
PROGR : pppppp   INPUT-OUTPUT ERROR : FILE ff   OP: oo
STATUS : nn
END OF RUN DUE TO PROVOKED ABEND
```

In most cases, examining the status and type of operation allows the user to find the cause of the abnormal end.

The following table contains the most common values for the status (STATUS) and type of operation (OP).

nn	STATUS
21	SEQUENCE ERROR
22	DUPLICATE KEY
23	NO RECORD FOUND
24	BOUNDARY VIOLATION (KSDS-RRDS)
30	SYSTEM ERROR
30	SYSTEM ERROR
34	BOUNDARY VIOLATION(SEQUENTIAL)
92	LOGIC ERROR (FOR EXAMPLE, OPENING AN ALREADY OPEN FILE)
93	FILE STILL OPEN UNDER CICS
95	INVALID OR INCOMPLETE FILE INFORMATION

oo	OPERATION
W	WRITE
RW	REWRITE
RU	READ UPD
OP	OPEN
CL	CLOSE
D	DELETE
R	READ
P	START
RN	READ NEXT

If this message is absent and the type of ABEND generated directly signals a problem in the VA Pac system programs, it will be necessary to contact the VA Pac technical support. Be sure to KEEP ALL LISTINGS that may be necessary to analyze the problem, for example:

-The report of the execution with // OPTION LOG,PARTDUMP
-The list of installed programs and files (Job 9INS)
-etc.

PROCEDURE RESUMING AFTER AN ABEND

If a procedure terminates for one of the following reasons:

- output file too small,
- partition too small to open a file,
- file open in another partition, or
- any error not specifically associated with processing programs or processed data,

after correcting the error, it is possible to restart the procedure from the point where it terminated.

This option saves processing time, but must be executed with care.

The JCL line '/* GOTO JCLSTxx' must be activated, without change of location, with the label of the step which produced the abend.

The following cautions must be observed:

- If the physical names of the temporary files are prefixed by the '%' character, the retrieval job must be submitted in the same partition as the initial job; or, the names of the temporary files must be substituted in the retrieval job, with the deletion of the '%' prefix and the addition of a suffix '.BG', '.F2', '.F3', etc. depending on whether the initial job was submitted by BG, F2, F3, etc.

- If the abend is due to an output file that is too small, make sure that a redefinition of the file won't affect the temporary files in the queue.

These files can be determined by referring to the order of execution in the description of the procedure and locating:

- the input files of the step where the abend occurred, and

- the input files of one of the subsequent steps, if they are not ouput from any of the steps to be resubmitted.

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: PTU001

INITIALIZATION OF FILES: PTAINI

```
.Input file
  PAC7MD : Physical Name = $XW..W$MODUL..INPUT

.Output files:
 -Table Descriptions File
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Table Contents File
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Output report:
 -Initialization review
  SYS005
```

5.4. EXECUTION JCL

```
===MOD INTA
* $X JOB JNM=$PRFJ.INTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.INTA        ****** $PROD $PREL ******
*  INITIALIZATION OF FILE
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$TRO.00TD.$SLIT
* $X SLI MEM=DF$TRO.00TV.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
Pactables 2.5                      E 12345678 ABC
/*
/. JCLST03
* --- STEP 03 --- PTAINI
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MD,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTAINI,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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 is run (in terms of number of transactions), it may be necessary to run 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
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Input transaction file (GETD or GETA output):
 -Update transactions
  PAC7MD:  Physical Name = $TXU..R$GETx..MD
  On Tape                 SYS013

.Output file
  PAC7TK : Physical Name = $XW..W$MODUL..TK

.Output report:
 -Input/output errors on files
  SYS005

.Sort.


PRINTING OF DESCRIPTIONS:  PTA290

.Permanent input file:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Transaction input file:
 -Print request
  PAC7TE : Physical Name = $XW..W$MODUL..TK

.Output report:
 -Printout of descriptions
  SYS005
```

6.3. EXECUTION JCL

```
===MOD GETT
* $X JOB JNM=$PRFJ.GETT,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.GETT        ****** $PROD $PREL ******
*  TABLE GENERATION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE TEMPORARY FILE     TK --- */
 DEL                ($XW..W$MODUL..TK) CL
 DEF CL       (NAME ($XW..W$MODUL..TK) -
               MODEL($MODELF) -
               REC   (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..TK.D) )
/*
/. JCLST02
* --- STEP 02 --- PTA250
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MD,'$PXU..RGETD.MD',,VSAM,DISP=(,KEEP,KEEP)
/* PAC7MD ON TAPE  : SYS013 ASSIGNED
/* ASSGN SYS013,TAPE
/* MTC REW,SYS013
/* TLBL PAC7MD
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TK,'$XW..W$MODUL..TK',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA250,SIZE=(AUTO,50K)
/. JCLST03
* --- STEP 03 --- PTA290
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$XW..W$MODUL..TK',,VSAM,DISP=(,DELETE,KEEP)
// EXEC PTA290,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..TK)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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,

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, PTA302, has been shipped since Pactables Rel. 2.0.

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

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

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

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: PTU001

TABLE UPDATE: PTA300

```
.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -User parameters file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Permanent input-output file:
 -Table contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Input transaction file:
 -Update transactions
  PAC7MS : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Print requests
  PAC7DE : Physical Name = $XW..W$MODUL..DE

.Output report:
 -Transaction review
  SYS005

.Work file:
 -Prepared transactions
  PAC7MT : Physical Name = $XW..W$MODUL..MT
```

FORMATTING FOR PRINTING: PTA350

```
.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Input transaction file:
 -Print request
  PAC7DE : Physical Name = $XW..W$MODUL..DE

.Output file:
 -Print file
  PAC7ET : Physical Name = $XW..W$MODUL..ET

.Output report:
 -Statistics on printing


  SYS005
```

PRINTING OF TABLES: PTA360

```
.Permanent inpput file:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Input Transaction file:
 -Print file
  PAC7ET : Physical Name = $XW..W$MODUL..ET

.Output report:
 -Printing of tables
  SYS005

.Sort.
```

7.4. EXECUTION JCL

```
===MOD UPTA
* $X JOB JNM=$PRFJ.UPTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.UPTA        ****** $PROD $PREL ******
*  TABLE UPDATE
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     DE --- */
 DEL                ($XW..W$MODUL..DE) CL
 DEF CL       (NAME ($XW..W$MODUL..DE) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..DE.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     ET --- */
 DEL                ($XW..W$MODUL..ET) CL
 DEF CL       (NAME ($XW..W$MODUL..ET) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(160 160) -
               RECFM(FB(160)) -
      ) DATA  (NAME ($XW..W$MODUL..ET.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     MT --- */
 DEL                ($XW..W$MODUL..MT) CL
 DEF CL       (NAME ($XW..W$MODUL..MT) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(095 095) -
               RECFM(FB(095)) -
      ) DATA  (NAME ($XW..W$MODUL..MT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTA300
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$XW..W$MODUL..DE',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7MS,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7MT,'$XW..W$MODUL..MT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA300,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTA350
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$XW..W$MODUL..DE',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7ET,'$XW..W$MODUL..ET',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA350,SIZE=AUTO
/. JCLST05
* --- STEP 05 --- PTA360
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7ET,'$XW..W$MODUL..ET',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA360,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..DE)    CL
 DEL                ($XW..W$MODUL..ET)    CL
 DEL                ($XW..W$MODUL..MT)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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 historical with action 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: PTU001

EXTRACTION OF TABLES FOR PRINTING: PTA320

```
.Permanent input files:
 -Table-description File
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -Table-contents File
  PAC7TV : Physical Name = $TXU..$TRO.00TV
 -User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Update transactions
  PAC7CA : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Print requests
  PAC7DE : Physical Name = $XW..W$MODUL..DE

.Output report:
 -Transaction review
  SYS005
```

PREPARATION FOR PRINTING: PTA350

```
.Permanent input files:
 -Table-description File
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Input transaction file:
 -Print requests
  PAC7DE : Physical Name = $XW..W$MODUL..DE

.Output file:
 -Print file
  PAC7ET : Physical Name = $XW..W$MODUL..ET

.Output report:
 -Statistics on printing
  SYS005
```

PRINTING OF TABLES: PTA360

```
.Permanent input file:
 -Tables-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Input transaction file:
 -Print file
  PAC7ET : Physical Name = $XW..W$MODUL..ET

.Output report:
 -Printing of tables
 SYS005

.Sort.
```

8.4. EXECUTION JCL

```
===MOD PRTA
* $X JOB JNM=$PRFJ.PRTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.PRTA        ****** $PROD $PREL ******
*  TABLE PRINTING
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     DE --- */
 DEL                ($XW..W$MODUL..DE) CL
 DEF CL       (NAME ($XW..W$MODUL..DE) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..DE.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     ET --- */
 DEL                ($XW..W$MODUL..ET) CL
 DEF CL       (NAME ($XW..W$MODUL..ET) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(160 160) -
               RECFM(FB(160)) -
      ) DATA  (NAME ($XW..W$MODUL..ET.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
LA
HA
SA
/*
/. JCLST03
* --- STEP 03 --- PTA320
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7CA,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7DE,'$XW..W$MODUL..DE',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA320,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTA350
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$XW..W$MODUL..DE',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7ET,'$XW..W$MODUL..ET',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA350,SIZE=AUTO
/. JCLST05
* --- STEP 05 --- PTA360
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7ET,'$XW..W$MODUL..ET',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA360,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..DE)    CL
 DEL                ($XW..W$MODUL..ET)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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.

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 and onwards.

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

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

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

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: PTU001


TABLE CHECK AND UPDATE:  PTA310

.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Permanent input-output file:
 -Table-contents files
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Input file:
 -Request transactions
  PAC7MV : Physical Name = $XW..W$MODUL..INPUT
 -External table-file
  PAC7NK : Physical Name = &TABX

.Output file:
 -Print requests
  PAC7DE : Physical Name = $XW..W$MODUL..DE

.Output report:
 -Execution report
  SYS005
```

FORMATTING OF PRINTOUT: PTA350

```
.Permanent input files:
 -Table-descriptions file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Input transaction file:
 -Print requests
  PAC7DE : Physical Name = $XW..W$MODUL..DE

.Output file:
 -Print file
  PAC7ET : Physical Name = $XW..W$MODUL..ET

.Output report:
 -Printing statistics


  SYS005
```

PRINTING: PTA360

```
.Permanent input file:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Input transaction file:
 -Print file
  PAC7ET : Physical Name = $XW..W$MODUL..ET

.Output report:
 -Table printout
  SYS005

.Sort.
```

9.4. EXECUTION JCL

```
===MOD IMTA
* $X JOB JNM=$PRFJ.IMTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.IMTA        ****** $PROD $PREL ******
*   IMPORT OF EXTERNAL TABLES
/* ************************************************************
/* SUBSTITUTIONS : £TABX EXTERNAL NAME OF FILE
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM TABX='£TABX'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     DE --- */
 DEL                ($XW..W$MODUL..DE) CL
 DEF CL       (NAME ($XW..W$MODUL..DE) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..DE.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     ET --- */
 DEL                ($XW..W$MODUL..ET) CL
 DEF CL       (NAME ($XW..W$MODUL..ET) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(160 160) -
               RECFM(FB(160)) -
      ) DATA  (NAME ($XW..W$MODUL..ET.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
 ATTTTTT
/*
/. JCLST03
* --- STEP 03 --- PTA310
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$XW..W$MODUL..DE',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7MV,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7NK,'&TABX',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA310,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTA350
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$XW..W$MODUL..DE',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7ET,'$XW..W$MODUL..ET',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA350,SIZE=AUTO
```

```
/. JCLST05
* --- STEP 05 --- PTA360
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7ET,'$XW..W$MODUL..ET',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA360,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..DE)    CL
 DEL                ($XW..W$MODUL..ET)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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 as options.

For user programs written in cobol II, RETA assigns an operational sign (+) to positive numeric data (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 administrator's 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	tttttt	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: PTU001

REORGANIZATION OF TABLE CONTENTS: PTA400

```
.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV
 -User parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Reorganization requests
  PAC7DR : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Reorganized-contents file
  PAC7TX : Physical Name = $TXU..R$MODUL..TX
 -Reorganized-table list file
  PAC7DE : Physical Name = $TXU..R$MODUL..DE
  On tape                  SYS011
```

NOTE: This file, whose description contains print requests, may be kept. Once the reorganization is finished, 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
  SYS005

.Sort.

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

VALIDIDATION OF TABLE CONTENTS: PTA410

```
.Updating input file :
 -Reorganization requests
  PAC7MB : Physical Name = $XW..W$MODUL..INPUT

.Input file:
 -Reorganized-contents file
  PAC7TX : Physical Name = $TXU..R$MODUL..TX

.Output file:
 -Validated-contents file
  PAC7TW : Physical Name = $TXU..R$MODUL..TW
  On tape                  SYS013

.Sort.
```

REORGANIZATION OF TABLE-DESCRIPTIONS: PTA420

```
.Permanent input file:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Input file:
 -Reorganized-table list file
  PAC7DE : Physical Name = $TXU..R$MODUL..DE

.Output files:
 -Reorganized table-description file
  PAC7TS : Physical Name = $TXU..R$MODUL..TS
  On tape                  SYS014
 -Table-description print request
  PAC7ML : Physical Name = $TXU..R$MODUL..ML
  On tape                  SYS013
```

NOTE: This file should 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.

.Sort.

BUILDING OF BACKUP FILE: PTA430

```
.Input files:
 -Validated-contents fille
  PAC7TW : Physical Name = $TXU..R$MODUL..TW
 -Reorganized-description file
  PAC7TS : Physical Name = $TXU..R$MODUL..TS
  On Tape                  SYS011

.Ouput file:
 -Backup file resulting from
  reorganization
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.NEWTC
```

TG FILE BACKUP: PTASVG

```
.Permanent input file:
 -User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG
.Output file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.NEWTC,DISP=OLD
```

10.4. EXECUTION JCL

```
===MOD RETA
* $X JOB JNM=$PRFJ.RETA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.RETA        ****** $PROD $PREL ******
*  TABLE REORGANIZATION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODELF) -
              REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE RESULT FILE      DE --- */
 DEL                ($TXU..R$MODUL..DE) CL
 DEF CL       (NAME ($TXU..R$MODUL..DE) -
              MODEL($MODELF) -
              REC  (500 500) -
              RECSZ(080 080) -
              RECFM(FB(080)) -
      ) DATA  (NAME ($TXU..R$MODUL..DE.D) )
    /* --- DELETE-DEFINE RESULT FILE      ML --- */
 DEL                ($TXU..R$MODUL..ML) CL
 DEF CL       (NAME ($TXU..R$MODUL..ML) -
              MODEL($MODELF) -
              REC  (500 1000) -
              RECSZ(080 080) -
              RECFM(FB(080)) -
      ) DATA  (NAME ($TXU..R$MODUL..ML.D) )
* $X SLI MEM=DF$TRO.00TC.$SLIT
    /* --- DELETE-DEFINE RESULT FILE      TS --- */
 DEL                ($TXU..R$MODUL..TS) CL
 DEF CL       (NAME ($TXU..R$MODUL..TS) -
              MODEL($MODELF) -
              REC  (1000 1000) -
              RECSZ(240 240) -
              RECFM(FB(240)) -
      ) DATA  (NAME ($TXU..R$MODUL..TS.D) )
    /* --- DELETE-DEFINE RESULT FILE      TW --- */
 DEL                ($TXU..R$MODUL..TW) CL
 DEF CL       (NAME ($TXU..R$MODUL..TW) -
              MODEL($MODELV) -
              REC  (5000 5000) -
              RECSZ(572 1071) -
      ) DATA  (NAME ($TXU..R$MODUL..TW.D) )
    /* --- DELETE-DEFINE RESULT FILE      TX --- */
 DEL                ($TXU..R$MODUL..TX) CL
 DEF CL       (NAME ($TXU..R$MODUL..TX) -
              MODEL($MODELV) -
              REC  (5000 5000) -
              RECSZ(572 1071) -
      ) DATA  (NAME ($TXU..R$MODUL..TX.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
GA
/*
```

```
/. JCLST03
* --- STEP 03 --- PTA400
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$TXU..R$MODUL..DE',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7DR,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// DLBL PAC7TX,'$TXU..R$MODUL..TX',,VSAM,DISP=(NEW,,DELETE)
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA400,SIZE=(AUTO,50K)
/. JCLST04
* --- STEP 04 --- PTA410
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TW,'$TXU..R$MODUL..TW',,VSAM
// DLBL PAC7TX,'$TXU..R$MODUL..TX',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA410,SIZE=(AUTO,50K)
/. JCLST05
* --- STEP 05 --- PTA420
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DE,'$TXU..R$MODUL..DE',,VSAM,DISP=(,,DELETE)
// DLBL PAC7ML,'$TXU..R$MODUL..ML',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TS,'$TXU..R$MODUL..TS',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA420,SIZE=(AUTO,50K)
/. JCLST06
* --- STEP 06 --- PTA430
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM
// DLBL PAC7TS,'$TXU..R$MODUL..TS',,VSAM
// DLBL PAC7TW,'$TXU..R$MODUL..TW',,VSAM
// EXEC PTA430,SIZE=AUTO
/. JCLST07
* --- STEP 07 --- PTASVG
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM,DISP=OLD
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// EXEC PTASVG,SIZE=AUTO
// GOTO JCLST08&TAPE
/. JCLST08
* --- STEP 08 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
/. STEPEND
* --- STEP 09 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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
  PAC7TD : Physical Name = $TXU..$TRO.00TD
.Output file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.NEWTC
```

TV BACKUP: PTASVV

```
.Permanent input file:
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV
.Output file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.NEWTC
```

TG BACKUP: PTASVG

```
.Permanent input file:
 -User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG
.Output file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.NEWTC
```

11.3. EXECUTION JCL

```
===MOD SVTA
* $X JOB JNM=$PRFJ.SVTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.SVTA        ****** $PROD $PREL ******
*  BACKUP TABLES
/* ************************************************************
/* SYMBOLICS:
/* TAPE    'N' (BACKUP ON DISK)
/*         'Y' (BACKUP ON TAPE)
/* TAPLAB  LABEL OF BACKUP TAPE
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$TRO.00TC.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTASVD
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTASVD,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- PTASVV
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM,DISP=OLD
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTASVV,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTASVG
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM,DISP=OLD
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTASVG,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
// GOTO JCLST05&TAPE
/. JCLST05Y
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL IN,'$TXU..SEQ.$TRO.00.NEWTC,,VSAM
// ASSGN SYS004,$TAP1
// PAUSE  MOUNT TAPE YM5 OVER 280 TO WRITE. THANK YOU.
// MTC REW,SYS004
// TLBL OUT
// ON $RC GT 0 GOTO $EOJ
// EXEC IDCAMS,SIZE=AUTO
 REPRO IFILE (IN) -
       OFILE (OUT ENV (PDEV (2400) RECFM(VB) RECSZ (1067) -
       BLKSZ(32760) NLBL))
/*
// ON $RC<16 CONTINUE
/. JCLST05N
/. STEPEND
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
// SETPARM TAPLAB='YM5'
```

12. PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)

12.1. INTRODUCTION

TRANSFER OF A DATABASE FROM ANOTHER PLATFORM

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

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

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

. Backup on the source site (SVTA procedure)

. Transfer of the TC file produced by SVTA onto the target platform,

. Retrieval of the file on the target platform (TCTA procedure),

. Restoration of the database (RSTA procedure), with, in input, the TC file built by the preceding step.

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

EXECUTION CONDITION

None. However, read the notes on the following page carefully.

USER INPUT

None.

NOTES

1. BACKUP TRANSFER

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

2. DISK SPACE

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

3. INITIAL AND RESULT FILES

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

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

12.2. DESCRIPTION OF STEPS

TCTA: DESCRIPTION OF STEPS

```
TC BACKUP SPLIT:  PTATC1

.Input backup file
  PAC7TC : Physical Name = &SAVIN

.Output work files:
 -Table-description sequential image
  PAC7SD : Physical Name = $XW..W$MODUL..SD
 -Table-contents sequential image
  PAC7SV : Physical Name = $XW..W$MODUL..SV
 -Parameter sequential image
  PAC7SG : Physical Name = $XW..W$MODUL..SG


TABLE-DESCRIPTION SORT:  PTATCD

.Input work file:
 -Table-description sequential image
  PAC7SD : Physical Name = $XW..W$MODUL..SD

.Output work file:
 -Sorted table descriptions
  PAC7AD : Physical Name = $XW..W$MODUL..AD

TABLE-CONTENTS SORT:  PTATCV

.Input work file:
 -Sequential image of table-contents
  PAC7SV : Physical Name = $XW..W$MODUL..SV

.Output work file:
 -Sorted table-contents
  PAC7AV : Physical Name = $XW..W$MODUL..AV


USER-PARAMETER SORT:  PTATCG

.Input work file:
 Sequential image of parameters
  PAC7SG : Physical Name = $XW..W$MODUL..SG

.Output work file:
 -Sorted user parameters
  PAC7AG : Physical Name = $XW..W$MODUL..AG


RECONSTITUTION OF THE TC BACKUP:  PTATC2

.Permanent output file:
 -TC backup in EBCDIC format
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.NEWTC
  On tape                  SYS014

.Input work files:
 -Table-description sequential image
  PAC7AD : Physical Name = $XW..W$MODUL..AD
 -Sequential image of contents
  PAC7AV : Physical Name = $XW..W$MODUL..AV
 -Sequential image of parameters
  PAC7AG : Physical Name = $XW..W$MODUL..AG
```

TC GENERATION SHIFT (+1): IDCAMS

This step performs the generation shift (+1) of TC file
according to the IDCAMS commands contained in the
SLI member RG$TRO.00TC

12.3. EXECUTION JCL

```
===MOD TCTA
* $X JOB JNM=$PRFJ.TCTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.TCTA        ****** $PROD $PREL ******
*  TRANSFER OF A PACTABLES DATABASE
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM SAVIN='£SAVIN'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE TEMPORARY FILE     AD --- */
 DEL                ($XW..W$MODUL..AD) CL
 DEF CL       (NAME ($XW..W$MODUL..AD) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(244 244) -
               RECFM(FB(244)) -
      ) DATA  (NAME ($XW..W$MODUL..AD.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     AG --- */
 DEL                ($XW..W$MODUL..AG) CL
 DEF CL       (NAME ($XW..W$MODUL..AG) -
               MODEL($MODELF) -
               REC  (100 100) -
               RECSZ(089 089) -
               RECFM(FB(089)) -
      ) DATA  (NAME ($XW..W$MODUL..AG.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     AV --- */
 DEL                ($XW..W$MODUL..AV) CL
 DEF CL       (NAME ($XW..W$MODUL..AV) -
               MODEL($MODELF) -
               REC  (5000 5000) -
               RECSZ(1063 1063) -
               RECFM(FB(1063)) -
      ) DATA  (NAME ($XW..W$MODUL..AV.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     SD --- */
 DEL                ($XW..W$MODUL..SD) CL
 DEF CL       (NAME ($XW..W$MODUL..SD) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(244 244) -
               RECFM(FB(244)) -
      ) DATA  (NAME ($XW..W$MODUL..SD.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     SG --- */
 DEL                ($XW..W$MODUL..SG) CL
 DEF CL       (NAME ($XW..W$MODUL..SG) -
               MODEL($MODELF) -
               REC  (100 100) -
               RECSZ(089 089) -
               RECFM(FB(089)) -
      ) DATA  (NAME ($XW..W$MODUL..SG.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     SV --- */
 DEL                ($XW..W$MODUL..SV) CL
 DEF CL       (NAME ($XW..W$MODUL..SV) -
               MODEL($MODELF) -
               REC  (5000 5000) -
               RECSZ(1063 1063) -
               RECFM(FB(1063)) -
      ) DATA  (NAME ($XW..W$MODUL..SV.D) )
* $X SLI MEM=DF$TRO.00TC.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTATC1
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7SD,'$XW..W$MODUL..SD',,VSAM
// DLBL PAC7SG,'$XW..W$MODUL..SG',,VSAM
// DLBL PAC7SV,'$XW..W$MODUL..SV',,VSAM
// DLBL PAC7TC,'&SAVIN',,VSAM
// EXEC PTATC1,SIZE=AUTO
/. JCLST03
* --- STEP 03 --- PTATCD
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7AD,'$XW..W$MODUL..AD',,VSAM
// DLBL PAC7SD,'$XW..W$MODUL..SD',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTATCD,SIZE=(AUTO,50K)
/. JCLST04
* --- STEP 04 --- PTATCV
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7AV,'$XW..W$MODUL..AV',,VSAM
// DLBL PAC7SV,'$XW..W$MODUL..SV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTATCV,SIZE=(AUTO,50K)
/. JCLST05
* --- STEP 05 --- PTATCG
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7AG,'$XW..W$MODUL..AG',,VSAM
// DLBL PAC7SG,'$XW..W$MODUL..SG',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTATCG,SIZE=(AUTO,50K)
/. JCLST06
* --- STEP 06 --- PTATC2
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7AD,'$XW..W$MODUL..AD',,VSAM
// DLBL PAC7AG,'$XW..W$MODUL..AG',,VSAM
// DLBL PAC7AV,'$XW..W$MODUL..AV',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM
// EXEC PTATC2,SIZE=AUTO
/. JCLST07
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
/. STEPEND
* --- STEP 08 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..AD)    CL
 DEL                ($XW..W$MODUL..AG)    CL
 DEL                ($XW..W$MODUL..AV)    CL
 DEL                ($XW..W$MODUL..SD)    CL
 DEL                ($XW..W$MODUL..SG)    CL
 DEL                ($XW..W$MODUL..SV)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

13. RESTORATION (RSTA)

13.1. 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.

13.2. DESCRIPTION OF STEPS

RSTA: DESCRIPTION OF STEPS

```
RESTORATION OF TD:  PTARSD

.Permanent output file:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Permanent input file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.TC
  On Tape                  SYS012


RESTORATION OF TV:  PTARSV

.Permanent output file:
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV

.Permanent input file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.TC
  On Tape                  SYS011

RESTORATION OF TG:  PTARSG

.Permanent output file:
 -User parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Permanent input file:
 -Table backup
  PAC7TC : Physical Name = $TXU..SEQ.$TRO.00.TC
  On Tape                  SYS011
```

13.3. EXECUTION JCL

```
===MOD RSTA
* $X JOB JNM=$PRFJ.RSTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.RSTA        ****** $PROD $PREL ******
*  TABLES RESTORATION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$TRO.00TD.$SLIT
* $X SLI MEM=DF$TRO.00TG.$SLIT
* $X SLI MEM=DF$TRO.00TV.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTARSD
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.TC',,VSAM
/* PAC7TC ON TAPE  : SYS012 ASSIGNED
/* ASSGN SYS012,TAPE
/* MTC REW,SYS012
/* TLBL PAC7TC
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PTARSD,SIZE=AUTO
/. JCLST03
* --- STEP 03 --- PTARSV
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.TC',,VSAM
/* PAC7TC ON TAPE  : SYS011 ASSIGNED
/* ASSGN SYS011,TAPE
/* MTC REW,SYS011
/* TLBL PAC7TC
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTARSV,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTARSG
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$TXU..SEQ.$TRO.00.TC',,VSAM
/* PAC7TC ON TAPE  : SYS011 ASSIGNED
/* ASSGN SYS011,TAPE
/* MTC REW,SYS011
/* TLBL PAC7TC
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// EXEC PTARSG,SIZE=AUTO
// EXEC LISTLOG
/&
* $X EOJ
```

14. LIST OF TABLE DESCRIPTIONS (LDTA)

14.1. 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.

14.2. 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	tttttt	Table number
23	8	MMDDCCYY	Historical account date

NOTE:

The input transactions are not validated; erroneous requests are not taken into account.

14.3. DESCRIPTION OF STEPS

LDTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TABLE-DESCRIPTION PRINTING: PTA290

```
.Permanent input file:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Input transaction file:
 -Print request
  PAC7TE : Physical Name = $XW..W$MODUL..INPUT

.Output report:
 -Table-description printout
  SYS005
```

14.4. EXECUTION JCL

```
===MOD LDTA
* $X JOB JNM=$PRFJ.LDTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.LDTA        ****** $PROD $PREL ******
*  LIST OF TABLE-DESCRIPTION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
 Z  TLS
 Z  TDS
/*
/. JCLST03
* --- STEP 03 --- PTA290
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTA290,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

15. PARAMETER UPDATE (PMTA)

15.1. 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.

15.2. 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 allowed.

15.3. DESCRIPTION OF STEPS

PMTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

PARAMETER UPDATE: PTA100

```
.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE

.Permanent input-output file:
 -User parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Extraction requests
  PAC7MV : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Parameter printing requests
  PAC7NU : Physical Name = $XW..W$MODUL..NU

.Output report:
 -Printing of descriptions
  SYS005
```

PRINTING OF USER PARAMETERS: PTA120

```
.Permanent input files:
 -Table description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -User parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Print requests
  PAC7NU : Physical Name = $XW..W$MODUL..NU

.Output report:
 -Printing of user parameters
  SYS005
```

15.4. EXECUTION JCL

```
===MOD PMTA
* $X JOB JNM=$PRFJ.PMTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.PMTA        ****** $PROD $PREL ******
*  USER PARAMETER UPDATE
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     NU --- */
 DEL                ($XW..W$MODUL..NU) CL
 DEF CL       (NAME ($XW..W$MODUL..NU) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..NU.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 USER    TAPASSWORD3
/*
/. JCLST03
* --- STEP 03 --- PTA100
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MV,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7NU,'$XW..W$MODUL..NU',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// EXEC PTA100,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTA120
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7NU,'$XW..W$MODUL..NU',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// EXEC PTA120,SIZE=AUTO
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..NU)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

16. TABLE EXTRACTION (EXTA)

16.1. INTRODUCTION

EXTA: TABLE EXTRACTION

INTRODUCTION

The EXTA procedure transforms table data into batch update transactions.

EXECUTION CONDITION

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

16.2. 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	'/'

16.3. DESCRIPTION OF STEPS

EXTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

EXTRACTION OF TABLE DATA: PTA150

```
.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -Table contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV
 -User parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Extraction requests
  PAC7MV : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Extracted transactions
  PAC7EX : Physical Name = $XW..W$MODUL..EX

.Output report:
 -Transaction review
  SYS005
```

PRINTING OF EXTRACTED TRANSACTIONS: PTA160

```
.Permanent input file:
 -Table description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD

.Input transaction file:
 -Extracted transactions
  PAC7EX : Physical Name = $XW..W$MODUL..EX

.Output report:
 -Printing of extracted data
  SYS005

.Output file:
 -Extracted transactions
  PAC7NU : Physical Name = $TXU..R$MODUL..NU
  On tape                  SYS013

.Sort.

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

16.4. EXECUTION JCL

```
===MOD EXTA
* $X JOB JNM=$PRFJ.EXTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.EXTA        ****** $PROD $PREL ******
*   TABLE EXTRACTION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     EX --- */
 DEL                ($XW..W$MODUL..EX) CL
 DEF CL       (NAME ($XW..W$MODUL..EX) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(120 120) -
               RECFM(FB(120)) -
      ) DATA  (NAME ($XW..W$MODUL..EX.D) )
    /* --- DELETE-DEFINE RESULT FILE      NU --- */
 DEL                ($TXU..R$MODUL..NU) CL
 DEF CL       (NAME ($TXU..R$MODUL..NU) -
               MODEL($MODELF) -
               REC  (1000 500) -
               RECSZ(80 32720) -
               RECFM(FB(80)) -
      ) DATA  (NAME ($TXU..R$MODUL..NU.D) BUFSP(65536) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
 ATTTTTT
/*
/. JCLST03
* --- STEP 03 --- PTA150
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7EX,'$XW..W$MODUL..EX',,VSAM
// DLBL PAC7MV,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PTA150,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTA160
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7EX,'$XW..W$MODUL..EX',,VSAM
// DLBL PAC7NU,'$TXU..R$MODUL..NU',,VSAM
/* PAC7NU ON TAPE  : SYS013 ASSIGNED
/* ASSGN SYS013,TAPE
/* MTC REW,SYS013
/* TLBL PAC7NU
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTA160,SIZE=(AUTO,50K)
/. STEPEND
```

```
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..EX)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

17. DIRECT CONSULTATION OF TABLES (TUTA)

17.1. INTRODUCTION

TUTA: DIRECT CONSULTATION OF TABLES

INTRODUCTION

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

17.2. 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.

17.3. DESCRIPTION OF STEPS

TUTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

DIRECT CONSULTATION OF TABLES: PTAU80

```
.Permanent input files:
 -Table-description file
  PAC7TD : Physical Name = $TXU..$TRO.00TD
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -Table-contents file
  PAC7TV : Physical Name = $TXU..$TRO.00TV
 -User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Request transactions
  PAC7MX : Physical Name = $XW..W$MODUL..INPUT

.Permanent output files:
 -Table-description file
  PAC7AD : Physical Name = $TXU..$TRO.00AD
 -Table-contents file
  PAC7AV : Physical Name = $TXU..$TRO.00AV

.Output report:
 -Transaction report
 SYS005

.Sort.
```

17.4. EXECUTION JCL

```
===MOD TUTA
* $X JOB JNM=$PRFJ.TUTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.TUTA        ****** $PROD $PREL ******
*  DIRECT CONSULTATION OF TABLES
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$TRO.00AD.$SLIT
* $X SLI MEM=DF$TRO.00AV.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTAU80
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7AD,'$TXU..$TRO.00AD',,VSAM
// DLBL PAC7AV,'$TXU..$TRO.00AV',,VSAM
// DLBL PAC7MX,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'$TXU..$TRO.00TV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTAU80,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

18. DISPATCHED TABLE MANAGEMENT (DTM OPTION)

18.1. 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 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.

18.2. 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.

18.3. DESCRIPTION OF STEPS (CDT1)

CDT1: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

CHECK OF TRANSACTIONS: PTAD05

```
.Permanent input files:
 -'Master' table-description file
  PAC7TD : Physical Name = &TDMAST
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
 -User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
 -Comparison request transactions
  PAC7MV : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Validated comparison request transactions
  PAC7MX : Physical Name = $XW..W$MODUL..MX

.Output report:
 -Transaction report
  SYS005
```

TABLE-DESCRIPTION COMPARISON AND EXTRACTION: PTAD10

```
.Permanent input files:
 -'Master' table-description file
  PAC7TD : Physical Name = &TDMAST
 -Error message file
  PAC7TE : Physical Name = $TXU..$TRO.00TE
 -'Slave' table-description file
  PAC7TS : Physical Name = &TDSLAV

.Input transaction file:
 -Validated transactions
  PAC7MX : Physical Name = $XW..W$MODUL..MX

.Output file:
 -Comparison result to be used as input of
  the CDT2 procedure
  PAC7TX : Physical Name = $TXU..R$MODUL..TX

.Output report:
 -Extraction printout
  SYS005
```

18.4. EXECUTION JCL

```
===SEQ FOR DTM
===MOD CDT1
* $X JOB JNM=$PRFJ.CDT1,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.CDT1        ****** $PROD $PREL ******
*  DISPATCHED TABLE MANAGEMENT : TABLE-DESCRIPTION COMPARISON
/* ************************************************************
/* SUBSTITUTION :
/* £TDMAST = NAME OF 'MASTER' FILE OF  TABLES-DESCRIPTION
/* £TDSLAV = NAME OF 'SLAVE'  FILE OF  TABLES-DESCRIPTION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM TDMAST='£TDMAST'
// SETPARM TDSLAV='£TDSLAV'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     MX --- */
 DEL                ($XW..W$MODUL..MX) CL
 DEF CL       (NAME ($XW..W$MODUL..MX) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..MX.D) )
    /* --- DELETE-DEFINE RESULT FILE      TX --- */
 DEL                ($TXU..R$MODUL..TX) CL
 DEF CL       (NAME ($TXU..R$MODUL..TX) -
               MODEL($MODELF) -
               REC  (1000 500) -
               RECSZ(240 240) -
               RECFM(FB(240)) -
      ) DATA  (NAME ($TXU..R$MODUL..TX.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
 A
/*
/. JCLST03
* --- STEP 03 --- PTAD05
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MV,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7MX,'$XW..W$MODUL..MX',,VSAM
// DLBL PAC7TD,'&TDMAST',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTAD05,SIZE=(AUTO,50K)
/. JCLST04
* --- STEP 04 --- PTAD10
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MX,'$XW..W$MODUL..MX',,VSAM
// DLBL PAC7TD,'&TDMAST',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TS,'&TDSLAV',,VSAM
// DLBL PAC7TX,'$TXU..R$MODUL..TX',,VSAM
// EXEC PTAD10,SIZE=AUTO
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..MX)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

18.5. 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
  PAC7TD : Physical Name = &TDSLAV
 -Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE

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

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


  PAC7TX : Physical Name = $TXU..RCDT1.TX

.Output report:
 -Update report
  SYS005

.Sort.
```

18.6. EXECUTION JCL (CDT2)

```
===SEQ FOR DTM
===MOD CDT2
* $X JOB JNM=$PRFJ.CDT2,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.CDT2        ****** $PROD $PREL ******
*  DISPATCHED TABLE MANAGEMENT : UPDATE TABLE-DESCRIPTION
/* ************************************************************
/* SUBSTITUTION :
/* £TDMAST = NAME OF 'MASTER' FILE OF  TABLES-DESCRIPTION
/* £TDSLAV = NAME OF 'SLAVE'  FILE OF  TABLES-DESCRIPTION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM TDSLAV='£TDSLAV'
// SETPARM TVSLAV='£TVSLAV'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- PTAD20
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TD,'&TDSLAV',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TV,'&TVSLAV',,VSAM
// DLBL PAC7TX,'$TXU..RCDT1.TX',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTAD20,SIZE=(AUTO,50K)
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

18.7. 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 Pactables' files. It can be executed even if the files are open to on-line use.

18.8. 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.

18.9. DESCRIPTION OF STEPS (CVTA)

CVTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TABLE-CONTENTS COMPARISON: PTAV10

```
.Permanent input files:
-Table-description file
  PAC7TD : Physical Name = &TD
-Error-message file
  PAC7TE : Physical Name = $TXY..$TRO.00TE
-Table-contents file
  PAC7TV : Physical Name = &TV
-User-parameter file
  PAC7TG : Physical Name = $TXU..$TRO.00TG

.Input transaction file:
-Comparison requests
  PAC7MV : Physical Name = $XW..W$MODUL..INPUT

.Output file:
-Comparison result
  PAC7EX : Physical Name = $XW..W$MODUL..EX

.Output report:
 -Transaction report
  SYS005
```

EXTRACTION OF UPDATE TRANSACTIONS: PTAV20

```
.Permanent input file:
-Table-Description file
  PAC7TD : Physical Name = &TD

.Input transaction file:
-Comparison result
  PAC7EX : Physical Name = $XW..W$MODUL..EX

.Output file:
-Update transactions for use as
 input of UPTA)
  PAC7NU : Physical Name = $TXU..R$MODUL..NU
  On tape                  SYS013

.Output report:
 -Printing of extracted transactions
  SYS005

.Sort.
```

18.10. EXECUTION JCL (CVTA)

```
===SEQ FOR DTM
===MOD CVTA
* $X JOB JNM=$PRFJ.CVTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.CVTA        ****** $PROD $PREL ******
*  DISPATCHED TABLE MANAGEMENT : TABLE-CONTENTS UPDATE
/* ************************************************************
/* SUBSTITUTION :
/* £TD = NAME OF 'MASTER' FILE OF  TABLES-DESCRIPTION
/* £TD = NAME OF 'SLAVE'  FILE OF  TABLES-DESCRIPTION
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM TD='£TD'
// SETPARM TV='£TV'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     EX --- */
 DEL                ($XW..W$MODUL..EX) CL
 DEF CL       (NAME ($XW..W$MODUL..EX) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(120 120) -
               RECFM(FB(120)) -
      ) DATA  (NAME ($XW..W$MODUL..EX.D) )
    /* --- DELETE-DEFINE RESULT FILE      NU --- */
 DEL                ($TXU..R$MODUL..NU) CL
 DEF CL       (NAME ($TXU..R$MODUL..NU) -
               MODEL($MODELF) -
               REC  (500 500) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($TXU..R$MODUL..NU.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
 *USER    PASSWORD
SA
/*
/. JCLST03
* --- STEP 03 --- PTAV10
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7EX,'$XW..W$MODUL..EX',,VSAM
// DLBL PAC7MV,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TD,'&TD',,VSAM
// DLBL PAC7TE,'$TXY..$TRO.00TE',,VSAM
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// DLBL PAC7TV,'&TV',,VSAM
// EXEC PTAV10,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTAV20
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7EX,'$XW..W$MODUL..EX',,VSAM
// DLBL PAC7NU,'$TXU..R$MODUL..NU',,VSAM
/* PAC7NU ON TAPE  : SYS013 ASSIGNED
/* ASSGN SYS013,TAPE
/* MTC REW,SYS013
/* TLBL PAC7NU
// DLBL PAC7TD,'&TD',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTAV20,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..EX)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

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

19.1. INTRODUCTION TO RETRIEVAL FROM RELEASE 7.3 (R3TA)

INTRODUCTION TO 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.

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

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

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

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

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

RESULT

Pactables files ready to be used in Release 2.5.

19.2. DESCRIPTION OF STEPS (R3TA)

R3TA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

BACKUP OF THE 7.3 TD FILE: PTAXVD

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

```
.Input file:
 -7.3 TD file
 PAC7TD : Physical Name = £OLDTD

.Output file:
 -Backed up 7.3 file
 PAC7TC : Physical Name = $XW..W$MODUL..TR
 On Tape                   SYS012
```

7.3 TV FILE BACKUP: PTAXVV

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

```
.Input file:
 -7.3 TV file
 PAC7TV : Physical Name = £OLDTV

.Output file:
 -7.3 backed up file
 PAC7TC : Physical Name = $XW..W$MODUL..TR
 On Tape                  SYS011
```

BACKUP OF 7.3 TG FILE: PTAXVG

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

```
.Input file:
 -7.3 TG file
 PAC7TG : Physical Name = £OLDTG

.Output file:
 -7.3 backed up file
 PAC7TC : Physical Name = $XW..W$MODUL..TR
 On Tape                  SYS011
```

CONVERSION OF 7.3 BACKUP INTO 2.5 BACKUP: PTAR20

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

```
.Input file:
 -7.3 backup file
 PAC7TC : Physical Name = $XW..W$MODUL..TR
 -User-parameter file
 PAC7MB : Physical Name = $XW..W$MODUL.INPUT

.Output file:
 -Temporary 2.5 backup
 PAC7TR : Physical Name = $TXU..SEQ.$TRO.00.NEWTC

.Output report:
 -Retrieval report
 SYS005
```

19.3. EXECUTION JCL (R3TA)

```
===MOD R3TA
* $X JOB JNM=$PRFJ.R3TA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.R3TA        ****** $PROD $PREL ******
*  RETRIEVAL FROM  7.3
/* ************************************************************
/* SUBSTITUTIONS :
/* £OLDTD  NAME OF PHYSICAL FILE TD OF 7.3
/* £OLDTG  NAME OF PHYSICAL FILE TG OF 7.3
/* £OLDTV  NAME OF PHYSICAL FILE TV OF 7.3
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM OLDTD='£OLDTD'
// SETPARM OLDTG='£OLDTG'
// SETPARM OLDTV='£OLDTV'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODELF) -
              REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     TC --- */
 DEL                ($XW..W$MODUL..TC) CL
 DEF CL       (NAME ($XW..W$MODUL..TC) -
              MODEL($MODELV) -
              REC  (5000 5000) -
              RECSZ(572 1071) -
      ) DATA  (NAME ($XW..W$MODUL..TC.D) )
* $X SLI MEM=DF$TRO.00TD.$SLIT
* $X SLI MEM=DF$TRO.00TG.$SLIT
* $X SLI MEM=DF$TRO.00TV.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
61
/*
/. JCLST03
* --- STEP 03 --- PTAXVD
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM
// DLBL PAC7TD,'&OLDTD',,VSAM,CAT=OLDCAT
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAXVD,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTAXVV
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM,DISP=OLD
// DLBL PAC7TV,'&OLDTV',,VSAM,CAT=OLDCAT
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAXVV,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- PTAXVG
```

```
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM,DISP=OLD
// DLBL PAC7TG,'&OLDTG',,VSAM,CAT=OLDCAT
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAXVG,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST06
* --- STEP 06 --- PTAR20
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM
// DLBL PAC7TR,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM,DISP=NEW
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAR20,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST07Y
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL IN,'$TXU..SEQ.$TRO.00.NEWTC,,VSAM
// ASSGN SYS004,$TAP1
// PAUSE  MOUNT TAPE YM5 OVER 280 TO WRITE. THANK YOU.
// MTC REW,SYS004
// TLBL OUT
// ON $RC GT 0 GOTO $EOJ
// EXEC IDCAMS,SIZE=AUTO
 REPRO IFILE (IN) -
       OFILE (OUT ENV (PDEV (2400) RECFM(VB) RECSZ (1067) -
       BLKSZ(32760) NLBL))
/*
// ON $RC<16 CONTINUE
/. JCLST07N
/. JCLST08
* --- STEP 08 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
/. STEPEND
* --- STEP 09 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
 DEL                ($XW..W$MODUL..TC)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
// SETPARM TAPLAB='YM5'
```

19.4. INTRODUCTION TO RETRIEVAL FROM RELEASE 7.2 (R2TA)

INTRODUCTION TO RETRIEVAL FROM RELEASE 7.2 (R2TA)

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

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

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

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

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

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

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

RESULT

Pactables files ready to be used in Release 2.5.

19.5. DESCRIPTION OF STEPS (R2TA)

R2TA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TRANSFORMATION OF TG INTO A SEQUENTIAL FILE: IDCAMS

```
.Input file
 -TG File
  PAC7IG : Physical name= &OLDTG

.Output file
 -Sequential TG file
  PAC7TG : Physical name= $XW..W$MODUL..AG
```

CONVERSION OF THE TG FILE FROM 7.2 TO 7.3: PTARTG

```
.Input file:
 -Sequential 7.2 TG file
  PAC7AG : Physical Name = $XW..W$MODUL..AG

.Output file:
 -7.3 TG file
  PAC7TG : Physical Name = $TXU..$TRO.00TG
 -Conversion report
 SYS005

.Input-output file:
 -7.2 TD file
  PAC7TD : Physical Name = £OLDTD
```

BACKUP OF TD FILE, REL. 7.3: PTAXVD

This step creates a backup of the 7.3 TD file.

```
.Input file:
 -7.2 TD file
  PAC7TD : Physical Name = £OLDTD

.Output file:
 -7.3 backup file
  PAC7TC : Physical Name = $XW..W$MODUL..TC
```

BACKUP OF TV FILE, REL. 7.3: PTAXVV

This step creates a backup of the 7.3 TV file.

```
.Input file:
 -7.2 TV file
  PAC7TV : Physical Name = £OLDTV

.Output file:
 -7.3 backup file
  PAC7TC : Physical Name = $XW..W$MODUL..TC
```

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
 PAC7TG : Physical Name = $TXU..$TRO.00TG

.Output file:
 -7.3 backup file
 PAC7TC : Physical Name = $XW..W$MODUL..TC
```

CONVERSION OF 7.3 BACKUP TO 2.5 BACKUP: PTAR20

This step creates a 2.5 backup from the 7.3 backup.

```
.Input file:
 -7.3 backup file
 PAC7TC : Physical Name = $XW..W$MODUL..TR
 -User-parameter file


 PAC7MB : Physical Name = $XW..W$MODUL..INPUT

.Output file:
 -Temporary 2.0 backup
 PAC7TR : Physical Name = $TXU..SEQ.$TRO.00.NEWTC
 On tape                  SYS013

.Output report:
 -Conversion report
 SYS005
```

TC GENERATION SHIFT (+1) : IDCAMS

This step performs the generation shift (+1) of the TC file according to the IDCAMS commands contained in the SLI member RG$TRO.00TC.$SLIT

19.6. EXECUTION JCL (R2TA)

```
===MOD R2TA
* $X JOB JNM=$PRFJ.R2TA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.R2TA        ****** $PROD $PREL ******
*  RETRIEVAL FROM 7.2
/* ************************************************************
/* SUBSTITUTIONS :
/* £OLDTD  NAME OF PHYSICAL FILE TD OF 7.2
/* £OLDTG  NAME OF PHYSICAL FILE TG OF 7.2
/* £OLDTV  NAME OF PHYSICAL FILE TV OF 7.2
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// SETPARM OLDTD='£OLDTD'
// SETPARM OLDTG='£OLDTG'
// SETPARM OLDTV='£OLDTV'
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     AG --- */
 DEL                ($XW..W$MODUL..AG) CL
 DEF CL       (NAME ($XW..W$MODUL..AG) -
               MODEL($MODELF) -
               REC  (1000 1000) -
               RECSZ(080 080) -
               RECFM(FB(080)) -
      ) DATA  (NAME ($XW..W$MODUL..AG.D) )
    /* --- DELETE-DEFINE TEMPORARY FILE     TC --- */
 DEL                ($XW..W$MODUL..TC) CL
 DEF CL       (NAME ($XW..W$MODUL..TC) -
               MODEL($MODELV) -
               REC  (5000 5000) -
               RECSZ(565 1065) -
      ) DATA  (NAME ($XW..W$MODUL..TC.D) )
* $X SLI MEM=DF$TRO.00TC.$SLIT
* $X SLI MEM=DF$TRO.00TG.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
61
/*
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7TG,'$XW..W$MODUL..AG',,VSAM
// DLBL PAC7IG,'&OLDTG',,VSAM,CAT=OLDCAT
// EXEC IDCAMS,SIZE=AUTO
 REPRO IFILE (PAC7IG) OFILE(PAC7TG)
/*
/. JCLST04
* --- STEP 04 --- PTARTG
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7AG,'$XW..W$MODUL..AG',,VSAM
// DLBL PAC7TD,'&OLDTD',,VSAM,CAT=OLDCAT
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTARTG,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- PTAXVD
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM
// DLBL PAC7TD,'&OLDTD',,VSAM,CAT=OLDCAT
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAXVD,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST06
* --- STEP 06 --- PTAXVV
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OLDCAT,'$VCAO',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM,DISP=OLD
// DLBL PAC7TV,'&OLDTV',,VSAM,CAT=OLDCAT
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAXVV,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST07
* --- STEP 07 --- PTAXVG
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM,DISP=OLD
// DLBL PAC7TG,'$TXU..$TRO.00TG',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAXVG,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST08
* --- STEP 08 --- PTAR20
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TC,'$XW..W$MODUL..TC',,VSAM
// DLBL PAC7TR,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM,DISP=NEW
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAR20,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST09Y
* --- STEP 09 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL IN,'$TXU..SEQ.$TRO.00.NEWTC,,VSAM
// ASSGN SYS004,$TAP1
// PAUSE  MOUNT TAPE YM5 OVER 280 TO WRITE. THANK YOU.
// MTC REW,SYS004
// TLBL OUT
// ON $RC GT 0 GOTO $EOJ
// EXEC IDCAMS,SIZE=AUTO
 REPRO IFILE (IN) -
       OFILE (OUT ENV (PDEV (2400) RECFM(VB) RECSZ (1067) -
       BLKSZ(32760) NLBL))
/*
// ON $RC<16 CONTINUE
/. JCLST09N
/. JCLST10
* --- STEP 10 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
/. STEPEND
* --- STEP 11 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL               ($XW..W$MODUL..INPUT) CL
 DEL               ($XW..W$MODUL..AG)    CL
 DEL               ($XW..W$MODUL..TC)    CL
```

```
/*
// EXEC LISTLOG
/&
* $X EOJ
// SETPARM TAPLAB='YM5'
```

20. RETRIEVAL PACTABLES 8.01

20.1. INTRODUCTION

RETRIEVAL PACTABLES 8.01 (RTTA) : PRESENTATION

The RTTA procedure retrieves Pactables files from releases 8.01 , making it possible to use the resulting files in Pactables Rel. 2.5.

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

Depending on the years present in the dates of the 8.01 release, the century digits are added by reference to a 'pivot' year specified on a parameter line provided as input to the procedure.

The procedure's test job is delivered with the default 'pivot' year '61' (which can be changed). This means that, for a date in Rel. 8.01 whose year is less than '61', the value '20' is assigned to the century. If the year is higher than 61, the value assigned to the century is '19'.

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

The retrieval of Pactables 8.01 includes the following steps:

. Backup in 8.01 format (SVTA procedure) of the TD, TV, and TG files of Release 8.01;

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

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

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

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

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

RESULT

Pactables files ready to be used in Release 2.5.

20.2. RTTA : ENTREES UTILISATEUR

RTTA: USER INPUT

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

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

20.3. RTTA: DESCRIPTION OF STEPS

RTTA: DESCRIPTION OF STEPS

```
BACKUP RETRIEVAL:  PTAR20

.Permanent input file:
 -8.01 backup file
  PAC7TC necessarily on tape : SYS012

.Permanent output file:
 -Temporary backup, rel. 2.5
  PAC7TR : Physical Name = $TXU..SEQ.$TRO.00.NEWTC

.Input file:
 -user parameter line


 PAC7MB : Physical Name = $XW..W$MODUL..INPUT

.Output report:
 -Retrieval report
 SYS005
```

TC GENERATION SHIFT (+1) : IDCAMS

This step performs the generation shift (+1) of the TC file according to the IDCAMS commands contained in the SLI member RG$TRO.00TC.$SLIT

20.4. RTTA: EXECUTION JCL

```
===MOD RTTA
* $X JOB JNM=$PRFJ.RTTA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.RTTA        ****** $PROD $PREL ******
*  RETRIEVAL FROM  8.XX
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$TRO.00TC.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
61
/*
/. JCLST03
* --- STEP 03 --- PTAR20
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TR,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM,DISP=NEW
// ON $RC GT 0 GOTO $EOJ
// EXEC PTAR20,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04Y
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL IN,'$TXU..SEQ.$TRO.00.NEWTC,,VSAM
// ASSGN SYS004,$TAP1
// PAUSE  MOUNT TAPE YM5 OVER 280 TO WRITE. THANK YOU.
// MTC REW,SYS004
// TLBL OUT
// ON $RC GT 0 GOTO $EOJ
// EXEC IDCAMS,SIZE=AUTO
 REPRO IFILE (IN) -
       OFILE (OUT ENV (PDEV (2400) RECFM(VB) RECSZ (1067) -
       BLKSZ(32760) NLBL))
/*
// ON $RC<16 CONTINUE
/. JCLST04N
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
```

```
* $X EOJ
// SETPARM TAPLAB='YM5'
```

21. INSTALLATION

21.1. INTRODUCTION

INSTALLATION

The installation procedure is divided into three main phases:

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

The Pactables function is installed with a special installation tape. The complete installation process is described in this chapter.

The user must be familiar with the PACTABLE technical char- acteristics which are described in this manual in order to properly prepare an environment for the installation proce- dure (disk space, VSAM catalogue and 'space', CICS options, etc.).

Once the environment is prepared, the installation can then take place as follows:

PREPARATION

Retrieval of the initial JCL from the tape and execution of this JCL.

. Backup of the installation tape,

. Copy of a complete JCL processing module,

. Retrieval of the complete PACTABLE installation and operation JCL.

INSTALLATION

. See Subchapter "INSTALLATION PROCESS".

TESTS
. On-line procedure tests,

. Batch procedure tests.

21.2. INSTALLATION TAPE

INSTALLATION TAPE

The installation tape (without labels) includes the following files:

RANK	RECORD	BLOCK	CONTENTS
1	80	80	Preparation initial JCL
2	80	80	Load-module of MM1JCL installation utility
3	80	3440	Pactables installation and operation JCL
4	90	32760	Pactables error message file
5	1067 (V)	32760	Backup of test tables
6	1136 (V)	32760	TUF work file
7-11			Backup of Pactables phases library

21.3. UNLOADING OF INITIAL JCL

UNLOADING OF INITIAL JCL

The first file on the installation tape, which contains the initial JCL, must be unloaded onto the on-site editor (LRECL 80 BLKSIZE 80).

```
* $$ JOB JNM=$JOB,CLASS=$CLASS,DISP=$DISP,USER='$USER'
* $$ LST CLASS=$PCLO,DISP=$PDIO
* $$ PUN CLASS=$PCLP,DISP=$PDIP,DEST=$DEST
// JOB $JOB    UNLOAD PACBASE INITIAL JCL
// ASSGN SYS004,TAPE
// PAUSE MOUNT PACBASE INSTALLATION TAPE ON SYS004
// UPSI 1
// EXEC DITTO
$$DITTO REW OUTPUT=SYS004
$$DITTO TC INPUT=SYS004,NFILES=1
$$DITTO RUN OUTPUT=SYS004
$$DITTO EOJ
/*
/&
* $$ EOJ
```

21.4. INITIAL JCL

INITIAL JCL

The function of the initial JCL is to generate the installation and operation JCL of the System.

This initial JCL is a parameterized skeleton which allows customization; it is processed by the MM1JCL utility program, which generates JCL that will be adapted according to the parameters specified by the user.

The unloading and execution of the MM1JCL utility is contained in the initial JCL, which includes three jobs:

.JOB0 Copies the installation tape onto a user tape; the latter is considered as the System back-up and must be used for the actual installation.

.JOB1 Unloads the MM1JCL and PACDEB utilities by LNKEDT. (through the RESTORE and LIBR functions).

.JOB2 Executes the MM1JCL utility, which creates the actual installation and operation JCL.

The MM1JCL output must be kept, since it may be used for reinstallations.

```
/* --------------------------------------------------------
/* I N S T A L L A T I O N    I N I T I A L      J C L
/* --------------------------------------------------------
/*
/* THIS JCL CONTAINS THREE JOBS :
/*    - COPY OF THE REFERENCE TAPE ON AN INSTALLATION
/*      TAPE,
/*    - RECEPTION OF THE INSTALLATION UTILITY IN A USER
/*      LIBRARY,
/*    - RECEPTION OF INSTALLATION AND OPERATION
/*      JCL UNDER TEXT EDITOR.
/*
/* THE VARIABLE ELEMENTS USED IN THESE THREE JOBS ARE :
/*    - A VSAM CATALOG FOR WORK FILES,
/*    - A LIBRARY FOR THE UTILITY,
/*    - THE POWER SPOOL FOR THE OUTPUT PUNCHED JCL,
/*    - A USER CODE FOR POWER,
/*    - TWO TAPE NAMES,
/*    - TWO TAPE UNITS ADRESSES.
/*
/* -------- REQUIRED SUBSTITUTIONS -------------------
/*
/* %USER = JOB IDENTIFIER (4 CHAR)
/* %ACCT = POWER ACCOUNTING INFORMATION (USER='%ACCT')
/*
/* %CATN = IDENTIFICATION OF VSAM CATALOG
/* %CATV = VSAM SPACE VOLUME SERIAL NUMBER
/*
/* %LIBN = IDENTIFICATION OF A LIBRARY FOR THE UTILITY
/* %LIB  = LIB FOR UTILITY
/* %SLIB = SUBLIB FOR UTILITY
/*
/* %PLST = PARAMETERS  * $$ LST
/* %PPCH = PARAMETERS  * $$ PUN  FOR RECEPTION UNDER TEXT EDITOR
/*
/* %INST = VOLUME SERIAL NUMBER OF REFERENCE TAPE
/* %BCKP = VOLUME SERIAL NUMBER OF BACK-UP TAPE
/*
/* %TAP1 = FIRST TAPE DRIVE ADDRESS
/* %TAP2 = SECOND TAPE DRIVE ADDRESS
```

```
/* *************************************************************
*         COPY OF THE INSTALLATION TAPE
/* *************************************************************
* $$ JOB JNM=%USERJOB0,CLASS=0,DISP=D,USER='%ACCT'
* $$ LST %PLST
// JOB JOB0         COPY OF THE INSTALLATION TAPE
// PAUSE MOUNT  : %INST,%TAP1,READ    %BCKP,%TAP2,WRITE
// UPSI 1
// EXEC DITTO
$$DITTO REW OUTPUT=%TAP1
$$DITTO REW OUTPUT=%TAP2
$$DITTO TT INPUT=%TAP1,OUTPUT=%TAP2,NFILES=11
$$DITTO RUN OUTPUT=%TAP1
$$DITTO EOJ
/*
/&
* $$ EOJ
```

```
* $$ JOB JNM=%USERJOB1,CLASS=0,DISP=D,USER='%ACCT'
* $$ LST %PLST
// JOB JOB1        COPY OF THE UTILITY
/* *************************************************************
*  *     LOADING OF "MM1JCL" UTILITY
/* *     IN THE PREDEFINED USER LIBRARY
/* *************************************************************
// PAUSE MOUNT THE INSTALLATION TAPE %BCKP ON %TAP2
```

```
 MTC REW,%TAP2
 MTC FSF,%TAP2,1
// ASSGN SYSIPT,%TAP2
// DLBL %LIB,'%LIBN',,VSAM
// LIBDEF PHASE,CATALOG=%LIB.%SLIB
// OPTION CATAL
 INCLUDE
// EXEC LNKEDT
/*
/&
* $$ EOJ




* $$ JOB JNM=%USERJOB2,CLASS=0,DISP=D,USER='%ACCT'
* $$ LST %PLST
* $$ PUN %PPCH
// JOB JOB2        CREATION OF INSTALLATION JCL
/* *************************************************************
*   --- CREATION OF THE INSTALLATION JCL BY THE MM1JCL UTILITY
/* *************************************************************
/*
/*  MODIFY THE PROVIDED LIST OF COMMANDS SPECIFYING THE INSTAL-
/*  LATION VARIANTS (IF ANY) BY REQUESTING, IF NECESSARY, THE
/*  SELECTION OF PORTIONS OF THE INSTALLATION JCL (JCL MODULES):
/*  ASSIGN THE APPROPRIATE VALUES TO THE INSTALLATION PARAMETERS
/*  AND, IF NEEDED, SPECIFY THE LINES TO BE ADDED AT THE BEGIN-
/*  NING AND THE END OF THE MODULES.
/*
/* *************************************************************
* --- STEP 1 --- IDCAMS DEFINE
// DLBL IJSYSUC,'%CATN',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL            (PTR.IMPL.PAC7GC) CL
  DEF CL   (NAME (PTR.IMPL.PAC7GC) -
           RUS NIXD CISZ(4096) -
           VOL (%CATV) RECSZ (104 104) TRK (10 5) )
  DEL            (PTR.IMPL.PAC7GI) CL
  DEF CL   (NAME (PTR.IMPL.PAC7GI) -
           RUS NIXD CISZ(4096) -
           VOL (%CATV) RECSZ (104 104) TRK (50 10) )
  DEL            (PTR.IMPL.PAC7LA) CL
  DEF CL   (NAME (PTR.IMPL.PAC7LA) -
           RUS NIXD CISZ(4096) -
           VOL (%CATV) RECSZ (137 137) TRK (5 1) )
  DEL            (PTR.IMPL.PAC7LI) CL
  DEF CL   (NAME (PTR.IMPL.PAC7LI) -
           RUS NIXD CISZ(4096) -
           VOL (%CATV) RECSZ (137 137) TRK (5 1) )
/*
* --- STEP 2 --- MM1JCL
// ASSGN SYS004,%TAP2
// PAUSE MOUNT THE INSTALLATION TAPE %BCKP ON %TAP2
// MTC REW,SYS004
// MTC FSF,SYS004,2
// ASSGN SYS005,SYSLST
// ASSGN SYS010,SYSIPT
// ASSGN SYS011,SYSPCH
// DLBL IJSYSUC,'%CATN',,VSAM
// DLBL %LIB,'%LIBN',,VSAM
// LIBDEF PHASE,SEARCH=%LIB.%SLIB
// DLBL WK1,'PTR.IMPL.PAC7GC',,VSAM
// DLBL WK2,'PTR.IMPL.PAC7GI',,VSAM
// DLBL LBI,'PTR.IMPL.PAC7LA',,VSAM
// DLBL LBO,'PTR.IMPL.PAC7LI',,VSAM
/*   THE GENERATED JCL IS PRODUCED AS PUNCHED OUTPUT
/*   THIS JCL IS TO BE RECEIVED UNDER A TEXT EDITOR
// EXEC MM1JCL,SIZE=AUTO
===SELL ENG              .LANGUAGE SELECTION
===IDENC CLIENT 'ANY CLIENT',SITE 'PROTOTYPE JCL'
===SELV DTM              .DISPATCHED TABLES MANAGEMENT
===PRM PCJ=0             .EXECUTION CLASS      (JOB CARD)
```

```
===PRM PCL=X             .OUTPUT CLASS         (LST CARD)
===PRM PCP=X             .OUTPUT CLASS         (PUN CARD)
===PRM PDJ=D             .JOBS DISPOSITION     (JOB CARD)
===PRM PDL=H             .OUTPUT DISPOSITION   (LST CARD)
===PRM PDP=H             .OUTPUT DISPOSITION   (PUN CARD)
===PRM PRFJ=%USER        .JOB NAME PREFIXES    (4 CHARACTERS)
===PRM PUJ='%ACCT'       .USER PARAMETER       (JOB CARD)
===PRM PWJ=''            .POWER JOB CARD ADDITIONAL PARAMETERS
===PRM PWL=',RBS=250'    .POWER LST CARD ADDITIONAL PARAMETERS
===PRM PWP=''            .POWER PUN CARD ADDITIONAL PARAMETERS
===PRM GROUP='PACT'      .CICS/RDO : PACTABLES RESOURCES GROUP
===PRM LIST='LCGI'       .CICS/RDO: PACBASE RESOURCE LIST
===PRM NLSR=1            .LSR POOL NUMBER
===PRM TAPEI=%BCKP       .INSTALLATION TAPE NAME
===PRM TAP1=%TAP1        .TAPE DRIVE ADDRESS
===PRM TAP2=%TAP2        .ALTERNATIVE TAPE DRIVE ADDRESS
===PRM VOLVWK=%CATV      .TEMPORARY/DYNMA. FILES VOLUME
===PRM VWORK=%CATV       .TEMPORARY FILES VOLUME
===PRM XW=%WPAC          .WORK FILES INDEX
===PRM WKBG1='12875,400' .WORK AREA 1/BG (400 TRACKS 3380)
===PRM WKF21='0,0'       .WORK AREA 1/F2  - OPTIONAL
===PRM WKF31='0,0'       .WORK AREA 1/F3       "
===PRM WKF41='11675,400' .WORK AREA 1/F4       "
===PRM WKF51='10475,400' .WORK AREA 1/F5       "
===PRM WKF61='0,0'       .WORK AREA 1/F6       "
===PRM WKF71='0,0'       .WORK AREA 1/F7       "
===PRM WKF81='0,0'       .WORK AREA 1/F8       "
===PRM WKF91='0,0'       .WORK AREA 1/F9       "
===PRM MODB='%LIBN'      .BATCH MODULES LIB PHYSICAL NAME
===PRM MODO='%LIBN'      .ONLINE MODULES LIB PHYSICAL NAME
===PRM SLIL='%LIBN'      .SLI MEMBERS LIB PHYSICAL NAME
===PRM PRLB='%LIBN'      .VSE PROCEDURES LIB PHYSICAL NAME
===PRM LMODB='%LIB'      .LIB / MODULES BATCH
===PRM LMODO='%LIB'      .LIB / MODULES TP
===PRM LSLI='%LIB'       .LIB / SLI MEMBER
===PRM LPRLB='%LIB'      .LIB / VSE PROCEDURES
===PRM SMODB='.%SLIB'    .SUBLIB / BATCH MODULES
===PRM SMODO='.%SLIB'    .SUBLIB / ON-LINE MODULES
===PRM SLIS='.%SLIB'     .SUBLIB / SLI MEMBER
===PRM SLIT='PAC'        .TYPE   / SLI MEMBER
===PRM SPRLB='.%SLIB'    .SUBLIB / VSE PROCEDURES
===PRM SORTWK='$$SORTWK' .DEFN SORTWORK SPACE PROCEDURE NAME
===PRM OPT='LOG,NODUMP'  .'// OPTION' CARD PARAMETERS
===PRM TRO=TF            .SYSTEM ROOT (2 CHARACTERS)
===PRM TXU='PT250.USR'   .INDEX OF USER FILES
===PRM TXY='PT250.SYS'   .INDEX OF PACTABLES SYSTEM FILES
===PRM VOLTUX=%CATV      .VOLUME OF INDEX COMPONENTS OF USER FILES
===PRM VOLTUD=%CATV      .VOLUME OF DATA COMPONENTS OF USER FILES
===PRM VOLTYX=%CATV      .VOLUME OF INDEX COMPONENTS OF SYST FILES
===PRM VOLTYD=%CATV      .VOLUME OF DATA COMPONENTS OF SYST FILES
===PRM VOLTSV=%CATV      .VOLUME OF JOURNAL/BACKUP FILES
===PRM VCAT='%CATN'      .PACTABLES VSAM CATALOG
===PRM VCAO='CATOLD'     .VSAM CATALOG FOR RETRIEVALS
===PRM MODELF='TAB250.MODELF' .FIXED MODEL SAM-ESDS FILE
===PRM MODELV='TAB250.MODELV' .VARIABLE MODEL SAM-ESDS FILE
===BEGMOD
./ ADD NAME=$TRO.00$ZMODUL
//%USER$ZMODUL JOB %ACCT,'VSE',CLASS=X,MSGCLASS=X,NOTIFY=%USER
/*XMIT A07POWER DLM=%%
===ENDMOD
%%
/*
* --- STEP 3 --- IDCAMS DELETE
// DLBL IJSYSUC,'%CATN',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL    (PTR.IMPL.PAC7GC) CL
  DEL    (PTR.IMPL.PAC7GI) CL
  DEL    (PTR.IMPL.PAC7LA) CL
  DEL    (PTR.IMPL.PAC7LI) CL
/*
/&
* $$ EOJ
```

21.5. COMPLETE JCL INSTALLATION

COMPLETE JCL INSTALLATION

The MM1JCL module reads the JCL skeleton file and outputs a JCL adapted by the user via commands which:

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

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

. Select the installation variants in order to generate the JCL needed for specific processing based on site and installation conditions.

. Add lines before and after the JCL modules in order to separate them.

This step can be executed as many times as necessary to generate a completely adequate JCL jobstream.

The MM1JCL provides a list of the newly created JCL by module taking into account all parameters and requested variants.

Once the JCL is created, it may be retrieved under a text editor. At this point, the actual installation procedure can begin.

It is advisable to create in the text editor library as many members as there are JCL modules, each module being a logical step in the installation procedure.

Adding the proper cards before and after the modules will automatically split up the JCL. (Refer to the examples in Subchapter "JCL MODULE SEPARATORS").

FILES USED BY MM1JCL

```
   Input      : JCL skeleton          : SYS004 (tape)
                Utility requests      : SYS010 (reader)
   Output     : Generated JCL         : SYS011 (punch)
                List of generated JCL : SYS005 (printer)
   Work files : WK1 (300,000 bytes), WK2 (300,000 bytes),
                LBI (Dummy) LBO (1 track)
```

MM1JCL INSTALLATION DEFAULT OPTIONS

```
. Variants   :  All.

. Parameters :  The values indicated are examples. They
                are not to be entered.

. JCL modules:  No selection, all modules are copied.
```

CODING OF MM1JCL COMMANDS

```
===SELL lll                 Selection of installation
                            language
                            lll = ENG  English
                                  FRA  French

===SELV vvvv                Variant selection
                            vvvv = variant code

===SELM mmmm1 mmmm2 ...     JCL module selection
                            mmmm1 = JCL module name
                            mmmm2 = JCL module name
                            etc.
                            If no ===SELM line is entered,
                            all the JCL modules are selected

===PRM  NNNN=pppp           Parameter
                            NNNN = parameter name
                            pppp = parameter value

NOTE:   On ===PRM or ===SELV lines, comments may be entered.
        They should be preceded by a period, without excee-
        ding column 72.

===BEGMOD         Insert lines at beginning of module
....1
....n             (Lines to insert in front of each module)

===ENDMOD         Insert lines at end of module
....1
....n             (Lines to insert after each module)
```

INSTALLATION DEFAULT SETTINGS

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

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

.PARAMETERS (===PRM):

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

.MODULES (===SELM):

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

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

A line: ./ ADD NAME=$MODULE

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

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

21.6. INSTALLATION VARIANTS

CHART OF VARIANTS

```
===SELV vvvv                  . Comment
```

vvvv	Meaning	Comment
DTM	Dispatched Table Manager	Extension

21.7. JCL MODULES

INSTALLATION JCL MODULES

JCL modules are identified by a four-character code. The modules used for the installation have a numeric prefix.

The ===SELM command of the MM1JCL utility enables you to select the modules to be unloaded. By default, all JCL modules are selected.

===SELM aaaa bbbb cccc

```
------------------------------------------------------------
! Code ! Contents                                ! Nature  !
!------!-----------------------------------------!---------!
! 1CSD ! CSD source: CICS tables (PCT,PPT)       ! CICS SYS!
! 1PCT ! Assembler source: transactions  (PCT)   ! CICS SYS!
! 1PPT ! Assembler source: programs        (PPT) ! CICS SYS!
! 1FCT ! Assembler source: files           (FCT) ! CICS SYS!
! 1SLI ! Cataloging of SLI members               ! VSE JCL !
! 1VSE ! Cataloging of VSE procedures            ! VSE JCL !
! 1PWR ! Complement to POWER submission JCL      ! VSE JCL !
!      !                                         !         !
! 2PHA ! Installation of executable modules      ! VSE JCL !
!      !                                         !         !
! 3SYS ! Unloading of error messages             ! VSE JCL !
!      !                                         !         !
! R2TA ! Retrieval of tables 7.2                 ! VSE JCL !
! R3TA ! Retrieval of tables 7.3                 ! VSE JCL !
! RTTA ! Retrieval of tables 8.xx                ! VSE JCL !
!      !                                         !         !
! 4ITB ! Initialization of TUF work file         ! VSE JCL !
! 5TTC ! Unloading of tests tables               ! VSE JCL !
!      !                                         !         !
! 6JCI ! Modifications of CICS JCL               ! VSE JCL !
!      !                                         !         !
! 9INS ! List of installed programs              ! VSE JCL !
! 9MVT ! Examples of batch transactions          ! Trans.  !
-----------------------------------------------------------!
```

OPERATION JCL MODULES:

Code	Contents	Nature
CDT1	Extract. of table description comparison	VSE JCL
CDT2	Table updating	VSE JCL
CVTA	Extract. of table contents comparison	VSE JCL
EXTA	Table extraction	VSE JCL
GETT	Table generation	VSE JCL
IMTA	Table import	VSE JCL
INTA	Initialization of table files	VSE JCL
LDTA	List of table descriptions	VSE JCL
PMTA	Parameter updating	VSE JCL
PRTA	Table printing	VSE JCL
RETA	Table reorganization	VSE JCL
RSTA	Table restoration	VSE JCL
SVTA	Table saving	VSE JCL
TCTA	Table transposition	VSE JCL
TUTA	Direct consultation of tables	VSE JCL
UPTA	Table updating	VSE JCL

21.8. JCL PARAMETERS

JCL PARAMETERS

```
SYNTAX:
===PRM PPPP=pppp                .Comments
```

- The parameter VALUES which include SPECIAL CHARACTERS must be entered between QUOTES (Example: '1000,600').
- COMMENTS on ===PRM lines must not exceed COLUMN 72.

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

Note: when the default value or the '<>' value is fille in, the parameter must be coded.

PARAMETERIZATION OF POWER CARDS

CODE	MEANING
PRFJ	4-character POWER job name prefix
PUJ	USER parameter of the * $$ JOB card
PCJ	CLASS parameter of the * $$ JOB card
PDJ	DISP parameter of the * $$ JOB card
PWJ	Supplementary parameter(s) of * $$ JOB card (Ex: PWJ = ',NTFY=USER1')
PCL	CLASS parameter of the * $$ LST cards
PDL	DISP parameter of the * $$ LST cards
PWL	Supplementary parameter(s) of * $$ LST card (Ex: PWL = ',RBS=100,DEST=RMTNODE')
PCP	CLASS parameter of the * $$ PUN cards (for retrieval under the editor)
PDP	DISP parameter of the * $$ PUN card (for retrieval under the editor)
PWP	Supplementary parameter(s) of * $$ PUN card (for retrieval under the editor)
LST2	LST parameter of the 2nd * $$ LST card
LST3	LST parameter of the 3rd * $$ LST card

PARAMETERS OF VSAM FILE NAMES

All Pactables files (except sortwork files) are VSAM files whose physical names are parameterized in the following format:

.TXU.TRO00ff	: Pactables User Files
.TXY.TRO00ff	: Pactables System Files
.TXU.SEQ.TRO00.OLDff	: Backup Files: generation (-1)
.TXU.SEQ.TRO00.ff	: generation (0)
.TXU.SEQ.TRO00.NEWff	: generation (+1)
.XW.WPROC.FF	: Work Files
.TXU.RPROC.ff	: Procedure Results Files

CODE	MEANING
TXU	Index of Pactables user files
TXY	Index of Pactables system files
TRO	Root of Pactables system (2 characters other than 99)
XW	Index of work file names (To be prefixed with '%' to permit simultaneous execution of procedures in two or more partitions: VSAM concatenates the partition's identifier and the physical name of the files which have this character in the first position.)
VCAT	VSAM catalogue of Pactables files (At installation time, all files are supposedly under the same catalog).

RESOURCE PARAMETERS (VOLUMES, DRIVES):

CODE	MEANING
TAPEI	Name of installation backup tape (used in the installation)
TAP1	Tape drive address
TAP2	Address of a second tape drive
VOLVWK	Volume with temporary VSAM work files
VOLTSV	Volume with backup files (To ensure data security, it is recommended to keep them separate.)
VOLTUD	Volume with DATA components from Pactables user files.
VOLTUX	Volume with INDEX components from Pactables system files.
VOLTYD	Volume with DATA components from Pactables system files.
VOLTYX	Volume with INDEX components from Pactables system files.
VWORK	Volume of NON-VSAM work files
WKBG1	3 work areas on the work unit, coded WORK.='p,n'
WKBG2	where: p = position of the 1st track
WKBG3	n = number of allocated tracks
WKFx1	Optional: Equivalent to the 3 preceding areas
WKFx2	to permit the concurrent execution of select
WKFx3	procedures in BG, F2, F3, etc.

PARAMETERS OF LOAD-MODULE LIBRARIES

CODE	MEANING
MODB	Physical name of the library (or CIL) of the product's batch programs
LMODB	Name of the product's program library
SMODB	Name of the product's batch program sub-library, preceded by a period: SMODB='.sublib'
MODO	Physical name of the library (or CIL) of the product's on-line programs.
SMODO	Name of the on-line program sub-library, preceded by a period: SMODO='.sublib'
PRLB	Physical name of the library (or PL) of VSE procedures
LPRLB	Name of the VSE procedure library
SPRLB	Name of the VSE procedure sub-library, preceded by a period: SPRLB='.sublib'
SLIL	Physical name of the library (or SL) of SLI members
LSLI	Name of the SLI-member library
SLIS	Name of the SLI-member sub-library, preceded by a period: SLIS='.sublib'
SLIT	Type of SLI members. Default: SLIT='PAC'

MISCELLANEOUS PARAMETERS

```
+-------+-----------------------------------------------+
! CODE  ! MEANING                                       !
+-------+-----------------------------------------------+
!       !                                               !
! OPT   ! Parameters of the card : //OPTION             !
!       ! Default : OPT=LOG, NODUMP                     !
! SORTWK! Name of the SORTWK procedure                  !
$SORTWK : Non-VSAM, default value        !
!       !   The procedures ($0SORTWK,$xSORTWK,...) are  !
!       !   provided in the '1VSE' module.              !
!       ! SORTWK=SORTWK   : SAM under VSAM Space Management!
!       !   The SORTWK procedure is provided in the '1VSE' !
!       !   module.                                     !
!       ! SORTWK=......   : Your own procedure          !
! LIST  ! CICS/RDO (Resource Definition Online):        !
!       !  List name for the package resources          !
! GROUP ! CICS/RDO (Resource Definition Online):        !
!       !  Group name for the installed product's resources!
+-------+-----------------------------------------------+
```

21.9. SEPARATION OF JCL MODULES

SEPARATION OF JCL MODULES

```
BEFORE AND AFTER LINES FOR JCL MODULES

===BEGMOD             ===ENDMOD
line 1                line 1
.....                 .....
line n                line n
```

The purpose of these commands is to make it easier to split the JCL file created by the MM1JCL utility into as many members as there are JCL modules.

The MM1JCL utility adds these lines before and after each JCL module, after replacing the symbolic parameters, if necessary.

The parameter $ZMODUL represents the module name; a period must be added after this parameter in order to concatenate it with a character string:

```
A$ZMODUL.B   gives  AXXXXB
A$ZMODUL..B  gives  AXXXX.B
```

In order to automatically split the JCL into CMS/XEDIT members:

1. Insert the following lines:

```
===BEGMOD
:READ xxxx$ZMODUL    (xxxx=up to 4-character prefix)
```

2. Execute the JCL processing job

3. Set the Punch received from POWER as the first card in the CMS reader (ORDER RDR nnnn where nnnn= spool number)

4. Adjust the Reader and Spool classes (SPOOL RDR CLASS *)

5. If the volume onto which the modules are loaded is to be shared among several machines, it must be accessed with filemode A (ACCESS cuu where cuu = disk address).

6. Activate automatic dispatching by providing the desired filetype (READ * filetype)

NOTE: You can give the filetype on the :READ card which you insert by the BEGMOD command as follows:

```
READ xxxx$ZMODUL filetype
```

7. Restore CMS initial configuration. (ACC 191 A then ACC cuu filemode)

To automatically split a SP2 or VSE/ESA library into members:

1. Insert the following cards:

```
===BEGMOD
CA xxxx$ZMODUL..P(xxxx=up to 4-character prefix)
===ENDMOD
/+
```

2. Execute the JCL processing job and store the MM1JCL PUNCH on a disk.

3. Set the PUNCH as input to LIBR with:

```
  ASSGN SYSIPT,DISK,.. (if the PUNCH is on a disk)
 // EXEC LIBR,PARM=' ACCESS SUBLIB=yourlib.sublib'
```

To automatically split a PDS into members in MVS:

1. Insert the following cards:

```
===BEGMOD
./ ADD NAME=$PRO$PFI$ZMODUL
//%USER$ZMODUL JOB
%ACCT,VSE,CLASS=X,NOTIFY=%USER
/*XMIT name_of_VSE_machine DLM=%%
===ENDMOD
%%
```

NOTE: This is also the JCL sample provided by default in step 2.

2. Run the JCL processing job.

3. Set the resulting PUNCH.

JCL modules are separated by the following command:
./ ADD NAME=<JCL_module>
This flow is the input to the IEBUPDTE used to catalog the various modules in the PDS provided for that purpose.

21.10. INSTALLATION PROCESS

INSTALLATION

Once all JCLs are obtained, Pactables installation comprizes the following steps:

1. Modification and submission of POWER JCL

2. Update of CICS tables
   2.1. CSD
   2.2. ASSEMBLER

3. Procedures cataloging
   3.1. SLI Members
   3.2. VSE Procedures

4. Installation of Pactables system
   4.1. Installation of executable modules
   4.2. Loading of error messages

5. Installation of test tables

6. Modification and submission of CICS JCL

7. List of installation programs

21.11. 1. CHANGING THE POWER JCL

MODIFICATIONS OF POWER JCL

The JCL module '1PWR' contains the lines to add in the POWER JCL submission in order to use the control card$ SLI cards).

```
===MOD 1PWR
/* *************************************************************
/*      --- MODIFICATIONS TO BE ADDED IN THE POWER JCL         *
* ==============================================================
*         ADD $LSLI$SLIS   IN LIBDEF SOURCE,SEARCH             *
* ==============================================================
/* *************************************************************
// DLBL $LSLI,'$SLIL',,VSAM
```

21.12. 2.1 UPDATING THE CICS TABLES (CSD)

CICS TABLE UPDATE (CSD)

CICS/RDO: The command lines of the DFHCSDUP utility in the 1CSD member enable you to update DFHCSD.

The files table is updated by the 1FCT member (see next subchapter).

The elements included in the 1CSD member are described in the next subchapter.

```
===MOD 1CSD
**********************************************
********** $PROD $PREL PROGRAMS **************
**********************************************
DEFINE PROGRAM($TRO.FT00) GROUP($GROUP)
DEFINE PROGRAM($TRO.FT90) GROUP($GROUP)
DEFINE PROGRAM($TRO.PLNK) GROUP($GROUP)
DEFINE PROGRAM($TRO.P500) GROUP($GROUP)
DEFINE PROGRAM($TRO.P510) GROUP($GROUP)
DEFINE PROGRAM($TRO.P512) GROUP($GROUP)
DEFINE PROGRAM($TRO.P520) GROUP($GROUP)
DEFINE PROGRAM($TRO.P522) GROUP($GROUP)
DEFINE PROGRAM($TRO.P530) GROUP($GROUP)
DEFINE PROGRAM($TRO.P540) GROUP($GROUP)
DEFINE PROGRAM($TRO.P550) GROUP($GROUP)
DEFINE PROGRAM($TRO.P560) GROUP($GROUP)
DEFINE PROGRAM($TRO.P570) GROUP($GROUP)
DEFINE PROGRAM($TRO.P580) GROUP($GROUP)
DEFINE PROGRAM($TRO.P590) GROUP($GROUP)
DEFINE PROGRAM($TRO.P599) GROUP($GROUP)
DEFINE PROGRAM($TRO.P600) GROUP($GROUP)
DEFINE PROGRAM($TRO.P610) GROUP($GROUP)
DEFINE PROGRAM($TRO.P620) GROUP($GROUP)
DEFINE PROGRAM($TRO.P820) GROUP($GROUP)
DEFINE PROGRAM($TRO.P920) GROUP($GROUP)
DEFINE PROGRAM($TRO.R980) GROUP($GROUP) RESIDENT(YES)
**********************************************
********** $PROD $PREL TRANSACTIONS **********
**********************************************
DEFINE TRANSACTION($TRO.00) GROUP($GROUP) PROGRAM($TRO.P500)
DEFINE TRANSACTION($TRO.01) GROUP($GROUP) PROGRAM($TRO.P510)
DEFINE TRANSACTION($TRO.02) GROUP($GROUP) PROGRAM($TRO.P520)
DEFINE TRANSACTION($TRO.03) GROUP($GROUP) PROGRAM($TRO.P530)
DEFINE TRANSACTION($TRO.04) GROUP($GROUP) PROGRAM($TRO.P540)
DEFINE TRANSACTION($TRO.05) GROUP($GROUP) PROGRAM($TRO.P550)
DEFINE TRANSACTION($TRO.06) GROUP($GROUP) PROGRAM($TRO.P560)
DEFINE TRANSACTION($TRO.07) GROUP($GROUP) PROGRAM($TRO.P570)
DEFINE TRANSACTION($TRO.08) GROUP($GROUP) PROGRAM($TRO.P580)
DEFINE TRANSACTION($TRO.09) GROUP($GROUP) PROGRAM($TRO.P590)
DEFINE TRANSACTION($TRO.90) GROUP($GROUP) PROGRAM($TRO.P600)
DEFINE TRANSACTION($TRO.91) GROUP($GROUP) PROGRAM($TRO.P610)
DEFINE TRANSACTION($TRO.92) GROUP($GROUP) PROGRAM($TRO.P620)
ADD GROUP($GROUP) LIST($LIST)
```

```
===MOD 1FCT
* **************************************************************
*            *** $PROD $PREL ***
*           FILE CONTROL TABLE : $TRO.XX
* **************************************************************
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TB,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(VARIABLE,BLOCKED),                 $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TD,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(FIXED,BLOCKED),                    $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TE,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(BROWSE)                            $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=2,BUFNI=1,STRNO=1,                    $Y.*
               RECFORM=(FIXED,BLOCKED),                    $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TG,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(FIXED,BLOCKED),                    $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TV,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(VARIABLE,BLOCKED),                 $Y.*
               FILSTAT=(ENABLED,OPENED)
```

21.13. 2.2 UPDATING THE CICS TABLES (ASSEMBLER)

UPDATE OF CICS TABLES (ASSEMBLER)

.PCT : Source module '1PCT' or '1CSD' ; transaction codes:

$TRO.00 (Option DTB=YES required), to access Pactables.

One transaction per database.

$TRO.90 to $TRO.92 (Option DTB=YES required), to manage installation parameters.

$TRO.01 to $TRO.09: internal transaction codes used by 'RETURN TRANSID'.

.PPT : Source module '1PPT' or '1CSD' ; Maps and Programs coded:

$TRO.Pxxx for programs
$TRO.Mxxx for maps,
where xxx is a numerical code
(See the full list in subchapter 'Libraries' in chapter 'INTRODUCTION').

One PPT for Pactables system.

.FCT : Source module '1FCT' ;

1 Pactables system file:

$TRO.00TE	: Error messages

3 user files:

$TRO.00TD	: Table description file,
$TRO.00TV	: Table content file,
$TRO.00TG	: User parameter file,
$TRO.00TB	: TUF-TP function's work file.

The option LOG=YES is required for TD,TV,TG and TB files.

The estimated number of INDEX, INDEX BUFFERS and DATA BUFFERS indicated are the minimum numbers required by the system.

The two characters '$PRO' of transaction codes, map and program names and file names are user-defined.

The $TRO.00 transaction enables you to perform current operations on the tables (consultation, updates).

The $TRO.90 transaction enables you to modify the installation parameters (language version, function keys) and the user parameters (passwords).

```
===MOD 1PCT
* **************************************************************
*            *** $PROD $PREL ***
*        PROGRAM CONTROL TABLE : $TRO.XX
* **************************************************************
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.00,                $Y.*
               PROGRAM=$TRO.P500,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.01,                $Y.*
               PROGRAM=$TRO.P510,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.02,                $Y.*
               PROGRAM=$TRO.P520,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.03,                $Y.*
               PROGRAM=$TRO.P530,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.04,                $Y.*
               PROGRAM=$TRO.P540,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.05,                $Y.*
               PROGRAM=$TRO.P550,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.06,                $Y.*
               PROGRAM=$TRO.P560,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.07,                $Y.*
               PROGRAM=$TRO.P570,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.08,                $Y.*
               PROGRAM=$TRO.P580,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.09,                $Y.*
               PROGRAM=$TRO.P590,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.90,                $Y.*
               PROGRAM=$TRO.P600,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.91,                $Y.*
               PROGRAM=$TRO.P610,DTB=YES
         DFHPCT TYPE=ENTRY,TRANSID=$TRO.92,                $Y.*
               PROGRAM=$TRO.P620,DTB=YES
```

```
===MOD 1PPT
* **************************************************************
*            *** $PROD $PREL ***
*       PROCESSING PROGRAM TABLE : $TRO.XX
* **************************************************************
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.FT00,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.FT90,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.PLNK,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P500,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P510,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P512,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P520,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P522,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P530,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P540,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P550,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P560,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P570,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P580,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P590,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P599,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P600,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P610,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P620,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P820,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.P920,PGMLANG=COBOL
 DFHPPT TYPE=ENTRY,PROGRAM=$TRO.R980,PGMLANG=COBOL
```

```
===MOD 1FCT
* **************************************************************
*            *** $PROD $PREL ***
*           FILE CONTROL TABLE : $TRO.XX
* **************************************************************
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TB,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(VARIABLE,BLOCKED),                 $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TD,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(FIXED,BLOCKED),                    $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TE,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(BROWSE)                            $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=2,BUFNI=1,STRNO=1,                    $Y.*
               RECFORM=(FIXED,BLOCKED),                    $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TG,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(FIXED,BLOCKED),                    $Y.*
               FILSTAT=(ENABLED,OPENED)
         DFHFCT TYPE=DATASET,DATASET=$TRO.00TV,            $Y.*
               ACCMETH=VSAM,                               $Y.*
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE),         $Y.*
               LSRPOOL=$NLSR,                              $Y.*
               BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
               LOG=YES,                                    $Y.*
               RECFORM=(VARIABLE,BLOCKED),                 $Y.*
               FILSTAT=(ENABLED,OPENED)
```

21.14. 3.1 CATALOGING THE SLI PROCEDURES

<u>$ SLI in POWER)</u>

The JCL Module '1SLI' contains the source of the procedures to be cataloged in a library which can be accessed in read-mode by POWER. (Card: // LIBDEF SOURCE,SEARCH in the POWER submission JCL).
The procedures are:

```
-  DBJB$TRO.00      Start of standard job

-  DF$TRO.00ff      DELETE-DEFINE of ff file

-  RG$TRO.00ff      Shift of the ff backup file
```

The 'DFxxxxff' members permit the centralized management of the parameters for allocating VSAM files within Pactables.

The 'RGxxxxff' members permit the centralized management of the number of backup file generations to be saved.

The 'DBJBxxxx' members permit the centralized management of the JCL common to all Pactables operation procedures that are used to start a job (such as allocating libraries).

```
===MOD 1SLI
* $X JOB JNM=$PRFJ.1SLI,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.1SLI        ****** $PROD $PREL ******
/* PROCEDURES CATALOGING
/* ************************************************************
// OPTION $OPT
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL $LSLI,'$SLIL',,VSAM
/* GOTO JCLSTXX
* ----- STEP 1 ----- LIBR
// EXEC LIBR,PARM=' AC S=$LSLI$SLIS'
CA DBJB$TRO.00.$SLIT R=Y
// ASSGN SYS005,SYSLST
// ASSGN SYS006,SYSRDR
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL $LMODB,'$MODB',,VSAM
// DLBL $LPRLB,'$PRLB',,VSAM
// LIBDEF PHASE,SEARCH=$LMODB$SMODB
// LIBDEF PROC,SEARCH=$LPRLB$SPRLB
/+
CA DFMODFBK.$SLIT R=Y
    /* --------- DELETE-DEFINE FIXBLK MODEL ------- */
 DEL                  ($MODELF) CL
 DEF CL         (NAME ($MODELF) -
                 VOL($VOLVWK) -
                 REC  (100 100) -
                 RECSZ(80 80) -
                 CISZ (8192) -
                 RECFM(FB(80)) -
                 RUS -
                 NAL -
                 NIXD -
      ) DATA    (NAME ($MODELF..D) )
/+
CA DFMODVBK.$SLIT R=Y
    /* --------- DELETE-DEFINE VARBLK MODEL ------- */
 DEL                  ($MODELV) CL
 DEF CL         (NAME ($MODELV) -
                 VOL($VOLVWK) -
                 REC  (100 100) -
                 RECSZ(80 200) -
                 CISZ (8192) -
                 RECFM(VB) -
                 RUS -
                 NAL -
                 NIXD -
      ) DATA    (NAME ($MODELV..D) )
/+
CA DF$TRO.00AD.$SLIT R=Y
    /* --- DELETE-DEFINE OF 00AD (TD IMAGE) FILE --- */
 DEL                  ($TXU..$TRO.00AD) CL
 DEF CL         (NAME ($TXU..$TRO.00AD) -
                 SHR(2 3) RUS KEYS(21 0) -
                 RECSZ(240 240) REC(1000 1000) -
      ) INDEX   (NAME ($TXU..$TRO.00AD.I) -
                 VOL($VOLTUX) -
      ) DATA    (NAME ($TXU..$TRO.00AD.D) -
                 VOL($VOLTUD) FSPC(10 5) CISZ(4096) )
/+
CA DF$TRO.00AV.$SLIT R=Y
    /* --- DELETE-DEFINE OF 00AV (TV IMAGE) FILE --- */
 DEL                  ($TXU..$TRO.00AV) CL
 DEF CL         (NAME ($TXU..$TRO.00AV) -
                 SHR(2 3) KEYS(35 4) -
                 RECSZ(1000 1100) REC(1000 1000) -
      ) INDEX   (NAME ($TXU..$TRO.00AV.I) -
                 CISZ(1024) VOL($VOLTUX) -
      ) DATA    (NAME ($TXU..$TRO.00AV.D) -
                 FSPC(10 5) CISZ(4096) VOL($VOLTUD) )
/+
CA DF$TRO.00TB.$SLIT R=Y
```

```
    /* --- DELETE-DEFINE OF 00TB  FILE --- */
 DEL                  ($TXU..$TRO.00TB) CL
 DEF CL         (NAME ($TXU..$TRO.00TB) -
                 SHR(2 3) KEYS(63 3) -
                 RECSZ(80 1140) REC(1000 1000) -
      ) INDEX   (NAME ($TXU..$TRO.00TB.I) -
                 CISZ(1024) VOL($VOLTUX) -
      ) DATA    (NAME ($TXU..$TRO.00TB.D) -
                 FSPC(10 5) CISZ(4096) VOL($VOLTUD) )
/+
CA DF$TRO.00TC.$SLIT R=Y
    /* --- DELETE-DEFINE OF TC FILE --- */
 DEL                  ($TXU..SEQ.$TRO.00.NEWTC) CL
 DEF CL         (NAME ($TXU..SEQ.$TRO.00.NEWTC) -
                 MODEL($MODELV) -
                 REC  (5000 5000) -
                 RECSZ(572 32760) -
                 BUFSP(65536) -
                 VOL  ($VOLTSV) -
      ) DATA    (NAME ($TXU..SEQ.$TRO.00.NEWTC.D) )
/+
CA DF$TRO.00TD.$SLIT R=Y
    /* --- DELETE-DEFINE OF 00TD  FILE --- */
 DEL                  ($TXU..$TRO.00TD) CL
 DEF CL         (NAME ($TXU..$TRO.00TD) -
                 SHR(2 3) RUS KEYS(21 0) -
                 RECSZ(240 240) REC(1000 1000) -
      ) INDEX   (NAME ($TXU..$TRO.00TD.I) -
                 VOL($VOLTUX) -
      ) DATA    (NAME ($TXU..$TRO.00TD.D) -
                 VOL($VOLTUD) FSPC(10 5) CISZ(4096) )
/+
CA DF$TRO.00TE.$SLIT R=Y
    /* --- DELETE DEFINE FICHIER 00TE --- */
 DEL                  ($TXY..$TRO.00TE) CL
 DEF CL         (NAME ($TXY..$TRO.00TE) -
                 SHR(2 3) KEYS(17 0)   -
                 RECSZ(90 90) REC(930 50) -
      ) INDEX   (NAME ($TXY..$TRO.00TE.I) -
                 CISZ(1024) VOL($VOLTYX)  -
      ) DATA    (NAME ($TXY..$TRO.00TE.D) -
                 FSPC(10 5) CISZ(4096) VOL($VOLTYD) )
/+
CA DF$TRO.00TG.$SLIT R=Y
    /* --- DELETE DEFINE OF 00TG FILE --- */
 DEL                  ($TXU..$TRO.00TG) CL
 DEF CL         (NAME ($TXU..$TRO.00TG) -
                 SHR(2 3) KEYS(22 0) -
                 RECSZ(85 85) REC(5000 5000) -
      ) INDEX   (NAME ($TXU..$TRO.00TG.I) -
                 CISZ(1024) VOL($VOLTUX) -
      ) DATA    (NAME ($TXU..$TRO.00TG.D) -
                 FSPC(10 5) CISZ(4096) VOL($VOLTUD) )
/+
CA DF$TRO.00TV.$SLIT R=Y
    /* --- DELETE-DEFINE OF 00TV  FILE --- */
 DEL                  ($TXU..$TRO.00TV) CL
 DEF CL         (NAME ($TXU..$TRO.00TV) -
                 SHR(2 3) KEYS(35 4) -
                 RECSZ(1000 1100) REC(1000 1000) -
      ) INDEX   (NAME ($TXU..$TRO.00TV.I) -
                 CISZ(1024) VOL($VOLTUX) -
      ) DATA    (NAME ($TXU..$TRO.00TV.D) -
                 FSPC(10 5) CISZ(4096) VOL($VOLTUD) )
/+
CA RG$TRO.00TC.$SLIT R=Y
    /* --- SHIFT OF BACKUP FILE        TC --- */
 DEL            ($TXU..SEQ.$TRO.00.OLDTC) CL
 ALTER           $TXU..SEQ.$TRO.00.TC -
 NEWNAME        ($TXU..SEQ.$TRO.00.OLDTC)
 ALTER           $TXU..SEQ.$TRO.00.TC.D -
 NEWNAME        ($TXU..SEQ.$TRO.00.OLDTC.D)
 ALTER           $TXU..SEQ.$TRO.00.NEWTC -
 NEWNAME        ($TXU..SEQ.$TRO.00.TC)
 ALTER           $TXU..SEQ.$TRO.00.NEWTC.D -
 NEWNAME        ($TXU..SEQ.$TRO.00.TC.D)
/+
/*
// EXEC LISTLOG
/&
* $X EOJ
```

21.15. 3.2 CATALOGING THE VSE PROCEDURES

CATALOGING OF VSE PROCEDURES

The JCL Module '1VSE' contains the source of the JCL procedures to catalog. These procedures are supplied as examples only and can be changed (for example, by altering the 'SORTWK' parameter of installation job 'JOB2').

The following VSE procedures are in the stream of procedures to catalog:

```
     SORTWK    Procedure whch allocates a SORTWK in SAM
               under VSAM

     $0SORTWK  Procedure which allocates a SORTWK in SAM
               for the BG partition

     $xSORTWK  Procedure which allocates a SORTWK in SAM
               for the Fx partition (F2 to F9)

     $0PACWK1  Procedures which allocate SAM work files
     $0PACWK2  for the BG partition

     $xPACWK1  Procedures which allocate SAM work files
     $xPACWK2  for the Fx partition (F2 to F9)
```

```
===MOD 1VSE
* $X JOB JNM=$PRFJ.1VSE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.1VSE        ****** $PROD $PREL ******
*          VSE PROCEDURES CATALOGING
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
* ----- STEP 01 ----- LIBR
// EXEC LIBR,PARM=' AC S=$LPRLB$SPRLB'
 CATALP SORTWK
// DLBL SORTWK1,'$XW..SORTWK',0,VSAM,DISP=(NEW,DELETE,DELETE),$Q
               RECSIZE=9999,RECORDS=(999,999)
/+
 CATALP $0SORTWK
// DLBL SORTWK1,'PACWK1.BG',0,DA
// EXTENT SYS001,$VWORK,,1,$WKBG1
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $2SORTWK
// DLBL SORTWK1,'PACWK1.F2',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF21
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $3SORTWK
// DLBL SORTWK1,'PACWK1.F3',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF31
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $4SORTWK
// DLBL SORTWK1,'PACWK1.F4',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF41
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $5SORTWK
// DLBL SORTWK1,'PACWK1.F5',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF51
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $6SORTWK
// DLBL SORTWK1,'PACWK1.F6',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF61
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $7SORTWK
// DLBL SORTWK1,'PACWK1.F7',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF71
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $8SORTWK
// DLBL SORTWK1,'PACWK1.F8',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF81
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
 CATALP $9SORTWK
// DLBL SORTWK1,'PACWK1.F9',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF91
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
/*
// EXEC LISTLOG
/&
* $X EOJ
```

21.16. 4.1 DEFINITION OF 'MODELs'

DEFINITION OF 'MODELs'

Execute job '2FIC'.

```
===MOD 2FIC
* $X JOB JNM=$PRFJ.2FIC,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.2FIC        ****** $PROD $PREL ******
*            DEFINITION OF 'MODELS'
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  LISTC ENT($MODELF)
  IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DFMODFBK.$SLIT
  END
  LISTC ENT($MODELV)
  IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DFMODVBK.$SLIT
  END
/*
// EXEC LISTLOG
/&
* $X EOJ
```

21.17. 4.2 INSTALLATION OF EXECUTABLE MODULES

INSTALLATION OF EXECUTABLE MODULES

Execute job '2PHA'.

```
===MOD 2PHA
* $X JOB JNM=$PRFJ.2PHA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.2PHA        ****** $PROD $PREL ******
*          INSTALLATION OF PACTABLES MODULES
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
// DLBL $LMODO,'$MODO',,VSAM
/* GOTO JCLSTXX
* --- STEP 1 --- LIBR
// ASSGN SYS004,$TAP1
// PAUSE MOUNT TAPE $TAPEI ON $TAP1, IN READ MODE
// MTC REW,SYS004
// MTC FSF,SYS004,6
// EXEC LIBR
 ACCESS SUBLIB=$LMODO$SMODO
 RENAME $TRO.*.PHASE:DT*.PHASE
RESTORE -
 CGI.LB.PACXDT.PHASE:$LMODB$SMODB -
===SEQ FOR DTM
 CGI.LB.PTAD05.PHASE:$LMODB$SMODB -
 CGI.LB.PTAD10.PHASE:$LMODB$SMODB -
 CGI.LB.PTAD20.PHASE:$LMODB$SMODB -
===SEQ
 CGI.LB.PTAINI.PHASE:$LMODB$SMODB -
 CGI.LB.PTARSD.PHASE:$LMODB$SMODB -
 CGI.LB.PTARSG.PHASE:$LMODB$SMODB -
 CGI.LB.PTARSV.PHASE:$LMODB$SMODB -
 CGI.LB.PTARTG.PHASE:$LMODB$SMODB -
 CGI.LB.PTAR20.PHASE:$LMODB$SMODB -
 CGI.LB.PTASVD.PHASE:$LMODB$SMODB -
 CGI.LB.PTASVG.PHASE:$LMODB$SMODB -
 CGI.LB.PTASVV.PHASE:$LMODB$SMODB -
 CGI.LB.PTATCD.PHASE:$LMODB$SMODB -
 CGI.LB.PTATCG.PHASE:$LMODB$SMODB -
 CGI.LB.PTATCV.PHASE:$LMODB$SMODB -
 CGI.LB.PTATC1.PHASE:$LMODB$SMODB -
 CGI.LB.PTATC2.PHASE:$LMODB$SMODB -
 CGI.LB.PTAU80.PHASE:$LMODB$SMODB -
===SEQ FOR DTM
 CGI.LB.PTAV10.PHASE:$LMODB$SMODB -
 CGI.LB.PTAV20.PHASE:$LMODB$SMODB -
===SEQ
 CGI.LB.PTAXVD.PHASE:$LMODB$SMODB -
 CGI.LB.PTAXVG.PHASE:$LMODB$SMODB -
 CGI.LB.PTAXVV.PHASE:$LMODB$SMODB -
 CGI.LB.PTA100.PHASE:$LMODB$SMODB -
 CGI.LB.PTA120.PHASE:$LMODB$SMODB -
 CGI.LB.PTA150.PHASE:$LMODB$SMODB -
 CGI.LB.PTA160.PHASE:$LMODB$SMODB -
 CGI.LB.PTA250.PHASE:$LMODB$SMODB -
 CGI.LB.PTA290.PHASE:$LMODB$SMODB -
 CGI.LB.PTA300.PHASE:$LMODB$SMODB -
 CGI.LB.PTA302.PHASE:$LMODB$SMODB -
 CGI.LB.PTA310.PHASE:$LMODB$SMODB -
 CGI.LB.PTA312.PHASE:$LMODB$SMODB -
 CGI.LB.PTA320.PHASE:$LMODB$SMODB -
 CGI.LB.PTA350.PHASE:$LMODB$SMODB -
 CGI.LB.PTA360.PHASE:$LMODB$SMODB -
 CGI.LB.PTA400.PHASE:$LMODB$SMODB -
 CGI.LB.PTA410.PHASE:$LMODB$SMODB -
 CGI.LB.PTA420.PHASE:$LMODB$SMODB -
 CGI.LB.PTA430.PHASE:$LMODB$SMODB -
 CGI.LB.PTA800.PHASE:$LMODB$SMODB -
 CGI.LB.PTA900.PHASE:$LMODB$SMODB -
 CGI.LB.PTU001.PHASE:$LMODB$SMODB -
 CGI.LB.DTFT00.PHASE:$LMODO$SMODO -
 CGI.LB.DTFT90.PHASE:$LMODO$SMODO -
 CGI.LB.DTPLNK.PHASE:$LMODO$SMODO -
 CGI.LB.DTP500.PHASE:$LMODO$SMODO -
 CGI.LB.DTP510.PHASE:$LMODO$SMODO -
 CGI.LB.DTP512.PHASE:$LMODO$SMODO -
 CGI.LB.DTP520.PHASE:$LMODO$SMODO -
 CGI.LB.DTP522.PHASE:$LMODO$SMODO -
 CGI.LB.DTP530.PHASE:$LMODO$SMODO -
 CGI.LB.DTP540.PHASE:$LMODO$SMODO -
 CGI.LB.DTP550.PHASE:$LMODO$SMODO -
 CGI.LB.DTP560.PHASE:$LMODO$SMODO -
 CGI.LB.DTP570.PHASE:$LMODO$SMODO -
 CGI.LB.DTP580.PHASE:$LMODO$SMODO -
 CGI.LB.DTP590.PHASE:$LMODO$SMODO -
 CGI.LB.DTP599.PHASE:$LMODO$SMODO -
 CGI.LB.DTP600.PHASE:$LMODO$SMODO -
 CGI.LB.DTP610.PHASE:$LMODO$SMODO -
 CGI.LB.DTP620.PHASE:$LMODO$SMODO -
 CGI.LB.DTP820.PHASE:$LMODO$SMODO -
 CGI.LB.DTP920.PHASE:$LMODO$SMODO -
 CGI.LB.DTR980.PHASE:$LMODO$SMODO -
 TAPE=SYS004 REPLACE=Y SCAN=NO
 ACCESS SUBLIB=LCO$TRO$SMODO
 RENAME DT*.PHASE:$TRO.*.PHASE
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

21.18. 4.3 LOADING OF ERROR MESSAGES

LOADING OF ERROR MESSAGES

The JCL Module '3SYS' contains the JCL to define and load the TE error message file.

```
===MOD 3SYS
* $X JOB JNM=$PRFJ.3SYS,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.3SYS        ****** $PROD $PREL ******
*    LOADING OF PACTABLES SYSTEM FILES
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$TRO.00TE.$SLIT
/*
/. JCLST02
* --- STEP 02 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL OUT,'$TXY..$TRO.00TE',,VSAM
// ASSGN SYS004,$TAP1
// PAUSE PUT INSTALLATION TAPE $TAPEI ON $TAP1
 MTC REW,SYS004
 MTC FSF,SYS004,3
// TLBL IN
// EXEC IDCAMS,SIZE=AUTO
 REPRO INFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (90) -
       BLKSZ($BLKTE)  NLBL)) -
       OUTFILE (OUT) RUS
/*
// EXEC LISTLOG
/&
* $X EOJ
```

21.19. 5. LOADING OF TEST DATABASE

LOADING OF TEST TABLES

The JCL Module '5TTC' loads the test database backup which is supplied on the installation tape. It must then be restore by the RSTA procedure.

This database is supplied for initial operations test only.

```
===MOD 5TTC
* $X JOB JNM=$PRFJ.5TTC,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ***************************************************************
// JOB $PRFJ.5TTC        ****** $PROD $PREL ******
*  LOADING OF TEST TABLES
/* ***************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE BACKUP FILE        TC --- */
 DEL                  ($TXU..SEQ.$TRO.00.NEWTC) CL
 DEF CL         (NAME ($TXU..SEQ.$TRO.00.NEWTC) -
                 MODEL($MODELV) -
                 REC  (5000 5000) -
                 RECSZ(568 1071) -
                 VOL  ($VOLTSV) -
      ) DATA    (NAME ($TXU..SEQ.$TRO.00.NEWTC.D) )
/*
/. JCLST02
* --- STEP 02 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// ASSGN SYS004,$TAP1
// PAUSE MONTER BANDE IMPLANTATION $TAPEI SUR $TAP1
 MTC REW,SYS004
 MTC FSF,SYS004,4
// TLBL IN
// DLBL OUT,'$TXU..SEQ.$TRO.00.NEWTC',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 REPRO INFILE (IN ENV (PDEV (2400) RECFM(VB) RECSZ (1067) -
       BLKSZ($BLKTC) NLBL)) -
       OUTFILE (OUT)
/*
/. STEPEND
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$TRO.00TC.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
```

21.20. 7. MODIFICATION OF THE CICS JCL

MODIFICATION OF CICS START-UP JCL

Once CICS tables have been bee updated and compiled and the files correctly defined, CICS start-up JCL must be completed with the JCL Module '6JCI', which contains:

```
- The DLBL of the Pactables On-Line Module library,
- The DLBL of the VSAM catalog used for Pactables,
- The DLBL of each table file:  TD, TV, TG, TB,
- The DLBL of system file: TE.
```

This module does NOT contain the //LIBDEF PHASE,SEARCH=.... card which is already included in the JCL. This card must be completed by the call of the .sublib Pactables On-Line Module library.

```
===MOD 6JCI
* ***************************************************************
*           ***  $PROD $PREL  ***
*           MODIFICATIONS TO PERFORM ON CICS JCL
*           =======================================
* ***************************************************************
*   ADD   $LMODO$SMODO  IN  LIBDEF PHASE , SEARCH
* ***************************************************************
// DLBL $LMODO,'$MODO',,VSAM
// DLBL TABCT,'$VCAT',,VSAM
// DLBL $TRO.00TD,'$TXU..$TRO.00TD',,VSAM,CAT=TABCT
// DLBL $TRO.00TE,'$TXY..$TRO.00TE',,VSAM,CAT=TABCT
// DLBL $TRO.00TG,'$TXU..$TRO.00TG',,VSAM,CAT=TABCT
// DLBL $TRO.00TV,'$TXU..$TRO.00TV',,VSAM,CAT=TABCT
// DLBL $TRO.00TB,'$TXU..$TRO.00TB',,VSAM,CAT=TABCT
```

21.21. 8. LIST OF INSTALLED PROGRAMS

```
LIST OF INSTALLED PROGRAMS AND FILES

Execute JCL module '9INS'.

This job prints:

- A list of batch and on-line programs, with their compi-
  lation date.
```

It is recommended that this list be kept, so that, in case of Pactables operating difficulties, the installation references can be sent to the Technical Support.

```
===MOD 9INS
* $X JOB JNM=$PRFJ.9INS,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* ************************************************************
// JOB $PRFJ.9INS        ****** $PROD $PREL ******
*     LIST OF INSTALLED MODULES
/* ************************************************************
/* PACXDT INPUT:
/*
/* .TO SELECT PROGRAMS, ONE LINE PER SELECTED PROGRAM, AS
/*  FOLLOWS:
/* - CHARACTERS 3-8 : CODE OF SELECTED PROGRAM
/*
/* .TO OBTAIN THE LIST OF ALL THE PROGRAMS, ONLY ONE INPUT
/*  LINE, AS FOLLOWS:
/* - CHARACTERS 3-4 : ROOT OF THE SYSTEM ($TRO)
/*
/* ************************************************************
// OPTION $OPT
* $X SLI MEM=DBJB$TRO.00.$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
    /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE   --- */
 DEL                ($XW..W$MODUL..INPUT) CL
 DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODELF) -
               REC  (1 200) -
      ) DATA  (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
  $TRO
/*
/. JCLST03
* --- STEP 03 --- PACXDT
// DLBL IJSYSUC,'$VCAT',,VSAM
// DLBL PAC7DT,'$TXU..$TRO.00TD',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PACXDT,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
 DEL                ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

21.22. UTILIZATION TESTS

TESTS

The tests include the following steps:

. Test on on-line use under CICS,
. Tests on batch updating, printing and reorganization,
. Tests on table generation.

The test deck contains three tables:

```
   .'ITEMS' and 'TEMPER' : without historical accounts.
   .'CUSTOM'             : with historical account.
```

On-line tests of the Pactables function:

Open Pactables test files under CICS.
Pass through screen branchings.
Execute some updates.

Batch tests:

The JCL module '9MVT' contains the test transactions.
Begin with the PMTA procedure.
The other tests can be executed in any order.

21.23. EXAMPLES OF TEST TRANSACTIONS

```
===MOD 9MVT
....+....1....+....2....+....3..EXAMPLES OF TRANSACTIONS FOR CDT1
 *********TEST
 ACUSTOM
 ATEMPER
....+....1....+....2....+....3..EXAMPLES OF TRANSACTIONS FOR CVTA
 *********TEST
SACUSTOM10031985    10031987
....+....1....+....2....+....3..EXAMPLES OF TRANSACTIONS FOR EXTA
 *USER1   USER1
 ACUSTOM10031985
....+....1....+....2....+....3..EXAMPLE OF TRANSACTION FOR INTA
C.G.I. ESSAI                        F 123456789ABC
....+....1....+....2....+....3..EXAMPLE OF TRANSACTION FOR PMTA
 ********TATEST
 ********TJ000100* $X JOB JNM=$PRFJ.PRTA
 USER1   TAUSER1   2
....+....1....+....2....+....3..EXAMPLES OF TRANSACTIONS FOR PRTA
 *********TEST
EACUSTOM10031985
EATEMPER
....+....1....+....2....+....3..EXAMPLE OF TRANSACTIONS FOR RETA
 *********TEST
GACUSTOM10031985
GATEMPER
....+....1....+....2....+....3..EXAMPLE OF TRANSACTIONS FOR TUTA
 *********TEST
 ACUSTOM10031985
 ATEMPER
....+....1....+....2....+....3..EXAMPLE OF TRANSACTIONS FOR UPTA
 *********TEST
 ACUSTOM10031985  *
AV 44190
 V 5555333***ATHENES
 V 6666333***MOSCOU
 V 8899000***PEKIN
AV 3333111
 V 6666111*MARAVEN*BOLIVAR*CARACAS*22300*VENEZUELA*3
 ATEMPER
 V GUAYAQUIL/0F 75 0C 24/0F 75 0C 24/0F 78 0C 25/0F 78 0C 25
 V-/0F 78 0C 25/0F 78 0C 25/0F 80 0C 27/0F 80 0C 27/0F 78 0C 25
 V-/0F 78 0C 25/0F 78 0C 25/0F 75 0C 24
 V PARIS/////0F 58 0C 14
 V FRANKFURT/0F 30 0C -1/0F 32 0C  0/0F 39 0C  4/0F 46 0C  7
 V-/0F 55 0C 13/0F 60 0C 15/0F 64 0C 18/0F 63 0C 17/0F 57 0C 14
 V-/0F 48 0C  9/0F 38 0C  4/0F 33 0C  1
```

AV FRANKFORT