



VisualAge Pacbase 2.5

**PACTABLES 2.5 – UNIX
OPERATIONS MANUAL**

DETIX000251A

Note

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

According to your license agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

<http://www.software.ibm.com/ad/vapacbase/support.htm>

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

First Edition (April 1999)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:

<http://www.software.ibm.com/ad/vapacbase/support.htm>

or to the following postal address:

IBM Paris Laboratory
VisualAge Pacbase Support
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1983, 1999. All rights reserved.

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

NOTICES

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

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

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

TRADEMARKS

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

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

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

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

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

TABLE OF CONTENTS

1. FOREWORD	9
2. PACTABLES COMPONENTS	11
2.1. INTRODUCTION	12
2.2. THE ON-LINE PROGRAM LIBRARY	13
2.3. THE BATCH PROGRAM LIBRARY	15
2.4. THE MACRO-STRUCTURES OF PUF-TP FACILITY	17
2.5. SYSTEM FILES	18
2.6. EVOLVING FILES	19
2.7. USER ACCESS FACILITIES	22
2.8. EXECUTION OF USER APPLICATIONS	23
3. ENVIRONMENT	24
3.1. INTRODUCTION	25
3.2. ARCHITECTURES	26
3.3. WORKSTATION	28
3.4. ON-LINE SERVER	29
3.5. BATCH SERVER	30
3.6. USER PARAMETERS TRANSACTION	32
4. MONITORS START-UP	33
4.1. TABTP: ON-LINE SERVER START-UP	34
4.2. TABPARM: PARM TRANSACTION START-UP	42
4.3. TABBATCH: BATCH SERVER START-UP	46
4.4. PAFLINK: UNIX DUMB TERMINAL START-UP	53
4.5. PAFLINK: WINDOWS DUMB-TERMINAL START-UP	56
5. THE BATCH PROCEDURES	57
5.1. INTRODUCTION	58
5.2. CLASSIFICATION OF PROCEDURES	59
5.3. STRUCTURE OF PROCEDURE COMMAND FILES	60
5.4. ADVICE ON USE	68
5.5. PROCEDURES START-UP	71
5.6. PRINT-OUTS PRODUCED	72
5.7. ABNORMAL EXECUTIONS	73
5.8. LIST OF "RUN TIME ERROR"	75
6. TABLE INITIALIZATION (INTA)	76
6.1. INTRODUCTION	77
6.2. USER INPUT	78
6.3. DESCRIPTION OF STEPS	79
6.4. EXECUTION JCL	80
7. TABLE GENERATION (GETT)	81
7.1. INTRODUCTION	82
7.2. DESCRIPTION OF STEPS	83
7.3. EXECUTION JCL	84
8. TABLE UPDATE (UPTA)	86
8.1. INTRODUCTION	87
8.2. USER INPUT	88
8.3. DESCRIPTION OF STEPS	89
8.4. EXECUTION JCL	91
9. TABLE PRINTING (PRTA)	93
9.1. INTRODUCTION	94

9.2. USER INPUT	95
9.3. DESCRIPTION OF STEPS.....	96
9.4. EXECUTION JCL.....	98
10. TABLE IMPORT (IMTA).....	100
10.1. INTRODUCTION	101
10.2. USER INPUT	103
10.3. DESCRIPTION OF STEPS.....	104
10.4. EXECUTION JCL.....	106
11. TABLE REORGANIZATION (RETA)	108
11.1. INTRODUCTION	109
11.2. USER INPUT	110
11.3. DESCRIPTION OF STEPS.....	111
11.4. EXECUTION JCL.....	113
12. BACKUP (SVTA).....	115
12.1. INTRODUCTION	116
12.2. DESCRIPTION OF STEPS.....	117
12.3. EXECUTION JCL.....	118
13. PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)	120
13.1. INTRODUCTION	121
13.2. DESCRIPTION OF STEPS.....	123
13.3. EXECUTION JCL.....	124
14. RESTORATION (RSTA)	127
14.1. INTRODUCTION	128
14.2. DESCRIPTION OF STEPS.....	129
14.3. EXECUTION JCL.....	130
15. LIST OF TABLE DESCRIPTIONS (LDTA).....	132
15.1. INTRODUCTION	133
15.2. USER INPUT	134
15.3. DESCRIPTION OF STEPS.....	135
15.4. EXECUTION JCL.....	136
16. PARAMETER UPDATE (PMTA).....	137
16.1. INTRODUCTION	138
16.2. USER INPUT	139
16.3. DESCRIPTION OF STEPS.....	141
16.4. EXECUTION JCL.....	142
17. TABLE EXTRACTION (EXTA).....	144
17.1. INTRODUCTION	145
17.2. USER INPUT	146
17.3. DESCRIPTION OF STEPS.....	147
17.4. EXECUTION JCL.....	148
18. DIRECT CONSULTATION OF TABLES (TUTA).....	150
18.1. INTRODUCTION	151
18.2. USER INPUT	152
18.3. DESCRIPTION OF STEPS.....	153
18.4. EXECUTION JCL.....	154
19. DISPATCHED TABLE MANAGEMENT (DTM OPTION)	155
19.1. TABLE DESCRIPTION COMPARISON (CDT1-CDT2).....	156
19.2. USER INPUT (CDT1).....	157
19.3. DESCRIPTION OF STEPS (CDT1).....	158
19.4. EXECUTION JCL (CDT1).....	159

19.5. DESCRIPTION OF STEPS (CDT2)	161
19.6. EXECUTION JCL (CDT2)	162
19.7. TABLE CONTENTS UPDATE (CVTA)	163
19.8. USER INPUT (CVTA)	164
19.9. DESCRIPTION OF STEPS (CVTA)	165
19.10. EXECUTION JCL (CVTA)	166
20. TABLE RETRIEVAL FROM RELEASES 8.XX OR 1.2	168
20.1. INTRODUCTION	169
20.2. RTTA : ENTREES UTILISATEUR	171
20.3. RTTA: DESCRIPTION OF STEPS	172
20.4. RTTA: EXECUTION JCL	173
21. COMPATIBILITY BETWEEN PACTABLES 2.5 AND VA PAC 1.6.....	174
22. INSTALLATION.....	176
22.1. INTRODUCTION	177
22.2. INSTALLING SERVERS	181
22.3. DESCRIPTION OF CREATED DIRECTORIES	187
22.4. INSTALLING TUF-TP MACRO-STRUCTURES	194
22.5. CREATING/DELETING SERVERS	195
22.6. CREATING NETWORK RESOURCES	199
22.7. INSTALLING WINDOWS DUMB-TERMINALS	202
22.8. INSTALLING UNIX DUMB-TERMINALS.....	204
22.9. UTILIZATION TESTS	205
22.10. MANAGING SEVERAL PACTABLES BASES	207
22.11. REINSTALLING PACTABLES.....	210

VisualAge Pacbase - Operations Manual	PAGE	9
TABLES - EXPLOITATION & INSTALLATION		
FOREWORD		1

1. FOREWORD

FOREWORD

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

HOW TO USE THIS MANUAL FOR SYSTEM INSTALLATION

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

- . The 2.5 Release is different from any former Pactables Release regarding installation parameters. The test case provided on the installation tape must be executed.
- . Once the installation is complete, read the chapter about the retrieval and follow the instructions carefully in order to ensure a thorough compatibility of existing data from the former release.

VisualAge Pacbase - Operations Manual	PAGE	11
TABLES - EXPLOITATION & INSTALLATION		
PACTABLES COMPONENTS		2

2. PACTABLES COMPONENTS

2.1. INTRODUCTION

INTRODUCTION

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

Two types of resources are therefore necessary:

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

NOTE:

The installation of the Pactables function is quite independent of that of other VisualAge Pacbase functions.

The implementation of the Pactables function requires data which must be defined and described with the VisualAge Pacbase Specifications Dictionary function. The Extraction Procedure required to operate the Pactables function is described in the VisualAge Pacbase 2.5 Operations Manual.

Options of the Pactables function are coded as follows:

- . Dispatched Table Management : DTM
- . Security System Interface : SEC (only with IBM MVS)

2.2. THE ON-LINE PROGRAM LIBRARY

PROGRAM CODE	OPERATION AND MEANING
D8T500	Access, FT or 'clear': initial screen
D8T510	C1 : mono-item consultation and CR, MO, AN update
D8T512	C1 : mono-item consultation and CR, MO, AN update for Rel. 2
D8T520	C : multi-item consultation and AN for deletion
D8T522	C : multi-item consultation and AN for deletion for Rel. 2
D8T530	LT : list of tables
D8T540	LS : list of sub-schemas/sub-systems
D8T550	LD : list of documentation
D8T560	C3 : consultation of an archived item
D8T570	'HELP' screen
D8T580	LH : list of historical accounts
D8T590	LJ, LE : print requests
D8T599	System error display
D8T600	Password and parameters update
D8T610	User code update
D8T620	Access authorization update
PTA800	Optimized access funtion
PTA802	Optimized access funtion
PTA900	Generalized access function
PTA902	Generalized access function
D8FT00	Tables extraction (TUF-TP FACILITY)
D8FT90	User Interface (TUF-TP FACILITY)

Size: approx. 2 mega-Bytes on HP9000
approx. 1 mega-Byte on DPX/2
approx. 1 mega-Byte on DPX/20
approx. 2 mega-Bytes on RS6000
approx. 1 mega-Byte on DEC 4000 (OSF1)
approx. 1 mega-Byte on TANDEM

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 Release 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

BATCH PROGRAMS

! CODE	! PROC.	! OPT.	! SIGNIFICANCE	!
! PTAINI	! INTA	!	! File Initialization	!
! PTARSD	! RSTA	!	! TD file restoration	!
! PTARSV	! -	!	! TV file restoration	!
! PTARSG	! -	!	! TG file restoration	!
! PTAR20	! RTTA	!	! Tables 8.2,1.2 --->2.0 retrieval	!
! PTASVD	! SVTA	!	! TD file backup	!
! PTASVV	! -	!	! TV file bcakup	!
! PTASVG	! -	!	! TG file backup	!
! PTAU80	! TUTA	!	! Direct consultation of tables	!
! PTA100	! PMTA	!	! Parameter update	!
! PTA120	! -	!	! - -	!
! PTA150	! EXTA	!	! Table extraction	!
! PTA160	! -	!	! - - -	!
! PTA250	! GETT	!	! Table generation	!
! PTA290	! -	!	! - - - / Lists	!
! -	! LDTA	!	! - - - / Lists	!
! PTA300	! UPTA	!	! Table updating	!
! PTA302	! -	!	!	!
! PTA310	! IMTA	!	! Table import	!
! PTA312	! -	!	!	!
! PTA320	! PRTA	!	! Table printing	!
! PTA350	! UPTA	!	!	!
! -	! IMTA	!	!	!
! -	! PRTA	!	!	!
! PTA360	! UPTA	!	!	!
! -	! IMTA	!	!	!
! -	! PRTA	!	!	!
! PTA400	! RETA	!	! Re-organizing tables	!
! PTA410	! -	!	! - -	!
! PTA420	! -	!	! - -	!
! PTA430	! -	!	!	!
! PTAD05	! CDT1	! DTM	! Table description comparison	!
! PTAD10	! -	! DTM	! - - -	!
! PTAD20	! CDT2	! DTM	! - - -	!
! PTAV10	! CVTA	! DTM	! Table data comparison	!
! PTAV20	! -	! DTM	! - - - - -	!

2.4. THE MACRO-STRUCTURES OF PUF-TP FACILITY

TUF-TP MACRO-STRUCTURES

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 in the "Method" sub-directory, under Pactables installation directory. 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. SYSTEM FILES

SYSTEM FILES

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

- . The executable batch and on-line modules

(installed in the \$TABDIR/batch/gnt and \$TABDIR/tp/gnt directories).

- . A file containing the PACTABLE error messages and HELP documentation.

```
.Size          : approx. 900 records
               : approx. 1,2 mega-Bytes
.Organization  : Indexed
.Ressize       : 90
.Key           : 17 (position 1)
.Utilization   : Batch and on-line
.Location      : $TABDIR/databases directory
.Internal Name : PAC7TE
```

Comment

\$TABDIR is an installation parameter.

2.6. EVOLVING FILES

EVOLVING FILES

Evolving files contain all user-entered data managed by the system either in on-line or batch modes.

The first two files constitute the Pactables files:

The table descriptions file (TD)

```
.Organization : Indexed
.Recsize      : 240
.Key          : 19 (position 1)
.Utilization  : Batch and on-line
.Location     : with the PACLAN/XSPECIFICATIONS
               DATABASE
.Internal name : PAC7TD
```

The table contents file (TV)

```
.Organization : Indexed
.Recsize      : 80 to 1100
.Key          : 33 (position 5)
.Utilization  : Batch and on-line
.Location     : $TABDIR/bases/"database_name"
.Internal name : PAC7TV
```

The third file contains user parameters which are necessary to the system. This file is managed by a specific batch procedure.

The user parameters file (TG)

```
.Organization : Indexed
.Recsize      : 85
.Key          : 22 (position 1)
.Utilization  : Batch and on-line
.Location     : $TABDIR/bases/"database_name"
.Internal name : PAC7TG
```

It includes user codes and access authorizations.

A fourth file contains a backup of the user files described below:

The backup (TC)

```
.Organization : Sequential
.Recsize      : 1061
.Utilization  : Batch
.Location     : $TABDIR/save/"database_name"
.Internal name : PAC7TC
```

Managing the TC backup

TC, the sequential backup is used on input and output of the batch procedures. Therefore, this file is created and used under two distinct names: TC when it is used on input and TC.NEW when it is created by the procedure.

At the end of successful execution each procedure which creates TC.NEW calls the commands file TCBACKIP.ini. This file is installed in the \$TABDIR/save/"database_name" and ensures by default a rotation of two copies: TC is renamed TC-1, TC.NEW is renamed TC and TC.NEW is deleted.

Users can modify the TCBACKUP.ini file if they want to adapt this standard management to their operation constraints.

COMMENTS ON THE LOCATION OF THE TD FILE

. Sites with VA Pac-Unix and Pactables

The TD file is located in the VA Pac-Unix specifications database. The VA Pac-Unix and Pactables installation procedures assign the TD file in this directory via the PAC7TD.ini files (located in the PACLAN/X and Pactables 'assign' directory).

Users wishing to move the TD file must modify the two PAC7TD.ini files.

. Sites with Pactables only

The TD file will be located with the other Pactables database files.

Comment:

\$TABDIR and "database_name" are installation parameters.

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

2.7. USER ACCESS FACILITIES

USER ACCESS FUNCTIONS

Some sub-programs enable user applications to access Pactables:

! CODE	! MEANING	!
! PTA800	! Optimized access function	!
! PTA802	! Generalized access function	!
! PTA900	! Generalized access function	!
! PTA902	! Optimized access function	!
! PTARSD	! Restoration of table descriptions	!
! PTARSV	! Restoration of table contents	!
! D8FT00	! TUF facility	!
! D8FT90	! TUF facility	!

These programs are available in two forms:

. '.gnt' compiled in the Pactables UNIX environment which can be executed with the same Micro Focus Run-Time

They are installed in the TABDIR/tp/gnt directory.

. SOURCE COBOL (.cbl) files, which can be compiled and used in a Micro Focus environment different from that of Pactables. They are installed in the \$TABDIR/cobol directory.

These sub-programs can be used in Batch or on-line user applications.

2.8. EXECUTION OF USER APPLICATIONS

EXECUTION OF USER APPLICATIONS

The user may want to execute applications in a Micro Focus environment different from that of Pactables, either in Windows, OS/2, another UNIX platform or on the same UNIX platform but with a different version of Micro Focus compiler; one that is incompatible with Pactables' compiler.

In this case the user must create an execution environment comprising:

- . user access function(refer to the previous subchapter),
- . Pactables TD and TV files used in the applications, which must be formed:
 - from a TC backup only containing the image of the TD and TV files (refer to the "Introduction" to the SVTA procedure),
 - with the help of the TD and TV files loading procedure, calling the PTARSD and PTARSV programs compiled and linked in the user Micro Focus environment.

TD and TV files loading procedure

Creating a commands file(.BAT on MS-DOS, .CMD on OS/2) as indicated in the "Introduction" to the RSTA procedure.

TD and TV files loading programs

The \$TABDIR/cobol directory contains the source COBOL files for the PTARSD and PTARSV programs.

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
ENVIRONMENT

PAGE 24

3

3. ENVIRONMENT

3.1. INTRODUCTION

INTRODUCTION

This chapter defines the environment and the resources necessary for operating Pactables on local PC network. It presents the different parts of Pactables (on-line servers, BATCH servers, file servers) and their components, their structure and their operating mode.

NOTE:

The Micro Focus COBOL Run-Time is required for the running of Pactables.

3.2. ARCHITECTURES

ARCHITECTURES

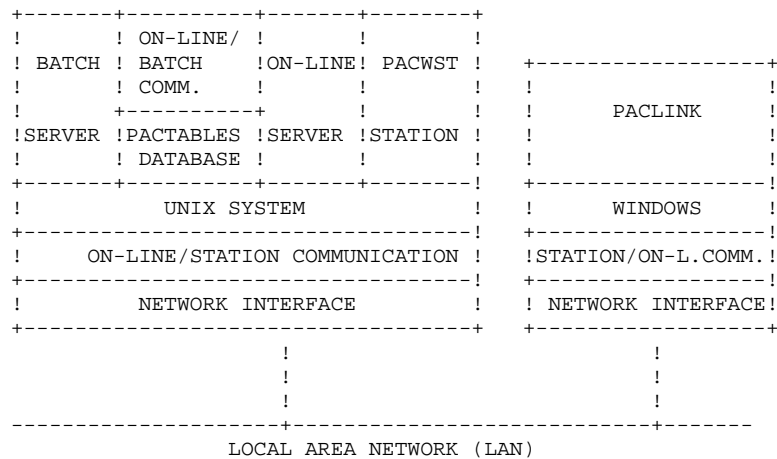
Each workstation communicates with one on-line server to access a Pactables database. An on-line server can in theory manage up to 99 stations, but for performance reasons it is better to limit the total number of workstations to 50.

It is possible to operate several on-line servers simultaneously on a UNIX machine.

Each workstation communicates with one or more BATCH servers to execute print requests - generation submitted on-line. There is no limit to the number of workstations a BATCH server can manage.

Resources allocation

The following diagram illustrates the allocation of programs, files and connections on the local network.



OPERATIONS PRINCIPLES

Communication must be established with the on-line server when using workstations.

The dialogue chosen to make the on-line server and its stations communicate is called 'process to process'. All information exchanges are made through the "socket" interface of the TCP/IP network protocol.

When a workstation submits a generation request, the on-line server communicates with the station in the usual way and transmits the print-generation requests to the BATCH server via a couple of communication files. The communication files between the on-line server and the BATCH server do not have to be shared on the local network.

3.3. WORKSTATION

WORKSTATION

Pactables' user environment includes WINDOWS workstations version 3.1 onwards, Windows 95 and Windows NT connected to the local network (PACLINK), and workstations running on the server of the UNIX system (TABWSTSOCK).

Each workstation communicates with the on-line server which executes programs, accesses Pactables database, and submits print-requests from the workstations to the batch server.

To enable communication between Windows workstations and the on-line server, a TCP/IP protocol, "Windows socket" compatible, must be installed on each workstation.

To be able to use Unix workstations (TABWSTSOCK), a Unix session must be opened on the station where the on-line server is defined.

The Pactables generation-print output files can be accessed, either by the share of the server's USERS directory, or by the use of a file transfer protocol (FTP for example).

3.4. ON-LINE SERVER

ON-LINE SERVER

Each on-line server is in charge of the execution of transactional programs allowing the Pactables database manipulation.

The dialog used to share information is carried out through a synchronous communication without using files.

To set a "socket" connection, you need to enter the hostname of the Unix machine supporting the on-line server and the "socket" number given once the server is created.

3.5. BATCH SERVER

BATCH SERVERS

The BATCH server's job is to process the print-generation requests submitted from TABLE's LE option screen. A BATCH monitor program ensures that these requests are taken into account. This program automatically submits the PRTA procedure.

The communication between the batch server and the on-line server(s) is carried out by the intermediary of the files defined in the BATCH server configuration file, often called "database_name".tlb and "database_name".tbd.

The display mode of the BATCH server ('view' option in tabbatch) allows the previous requests and those being executed to be consulted. Each request is numbered and processed by the monitor sequentially.

It is possible to install several BATCH servers processing the print-generation requests for a TABLE database. In this case, the BATCH servers, each with a different name, communicate with the on-line server(s) by the same pair of TLB and TBD files and share the execution of the print-generation requests.

For more details, refer to chapter "MONITORS START-UP, subchapter "BATCH SERVER START-UP".

BATCH IN OPERATION

The principle of BATCH is similar to that of on-line; it is based on the exchange of messages between an on-line server and a BATCH server.

The TLB contains a list of requests from the workstations LE screen.

The BATCH monitor reads the TLB file.

For each request TLB contains:

- the date and time of submission,
- the job number,
- the user code,
- the database code,
- the address of the start of the request in the TBD file,
- the address of the end of the request in the TBD file.

The TBD file, pointed to by TLB, contains the print-generation commands.

The result of PRTA is a generation and print type file. These files are created in the users directory (refer to chapter "INSTALLATION", subchapter "DESCRIPTION OF CREATED DIRECTORIES").

3.6. USER PARAMETERS TRANSACTION

THE TABPARM TRANSACTION

The TABPARM transaction allows the Pactables database administrator to update the user parameters interactively.

The tabparm program starts up this transaction.

This program is for a single user and cannot be accessed from a workstation. Therefore it must be started up on the UNIX machine; the dialog is carried out directly on the UNIX terminal.

For more details refer to chapter "MONITORS START-UP", subchapter "TABPARM TRANSACTION START-UP".

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
MONITORS START-UP

PAGE 33

4

4. MONITORS START-UP

4.1. TABTP: ON-LINE SERVER START-UP

ON-LINE MONITOR START-UP

The on-line monitor, whose executable programs (tabtp, tabserver, tablaunch) are in the TABDIR/bin directory, allows for:

- making the on-line server(s) active or inactive,
- supplying information concerning the on-line server(s),
- purging a station or stations linked to an on-line server,
- purging the on-line server(s).

To carry out the operations above the on-line monitor interpreter commands (tabtp) are executed. There are two possible modes:

- the command mode,
- the "shell" mode.

The command mode consists in typing:

```
tabtp <command>
```

To pass to "shell" mode type:

```
tabtp -s
```

The advantage of "command mode" is that the on-line monitor commands can be inserted into the commands file.

For example, the com_table file contains the following commands:

```
display the servers reports
tabtp info
start-up the tp1 on-line server
tabtp start tp1
display the information on tp1 server for ten seconds
tabtp info tp1
sleep 10
stop the tp2 on-line server
tabtp stop tp2
start up the 003 station purging command
depending on the tp1 server
tabtp purge tp1 003
```

The advantage of the "shell" mode there is no need to restart the interpreter for each command as the latter waits for the next command.

The commands available are as follows:

- debug (activation/deactivation of the debugger),
- exit (exit "shell" mode),
- help (help on the systax of a command),
- info (information on the server(s),
- purge (purge a station),
- purge_server (purge the on-line server),
- shutdown (stop the server without confirmation),
- start (start the server),
- stop (stop the server with confirmation),
- trace (activation/deactivation of the trace),
- view (display the server's report).

The detailed description of the commands can be found below in 'Commands' Description' paragraph.

EXECUTION CONDITIONS

The TABDIR environment variable must be initialized (refer to chapter "INSTALLATION").

The COBPATH environment variable must contain the access path to the "\$TABDIR/tp/gnt" on-line modules.

CONFIGURATION OF ON-LINE SERVER

Each time a server is started-up, it executes a command file of "server_name".ini type to assign the environment variables that it needs.

This file is created in the TABDIR/assign/tp_server directory when the server is created.

. On-line server environment variables:

- BASENAME : database name
- RADICAL : COBOL programs' prefix
- PAC7LB et PAC7BD : BATCH communication files
- STATIONS : maximum number of workstations
- PACPAUSE : time of workstation interrogations
in milli-seconds
- PACSOCKET : port number (socket)

. Database environment variables:

- PAC7TD : tables descriptions file
- PAC7TE : error messages file
- PAC7TV : tables contents file
- PAC7TG : user parameters file

NOTE:

It is strongly advised to use the tabadmin procedure to create and delete on-line servers.

DESCRIPTION OF COMMANDS

Debug command

This command activates and deactivates the "debug" mode for the on-line server. The server's name must be passed as a parameter, followed by "on" or "off" to activate or deactivate the "debug" mode.

A resulting file is updated. It is called:
"server_name_process_number".SPY

and is found in the \$TABDIR/log directory.

EXAMPLE: start-up the "debug" mode on an on-line server called tp1.

```
tabtp debug tp1 on      (in command mode)
Pactables: debug tp1 on (in "shell" mode)
```

EXAMPLE: stop the "debug" mode on a tp1 on-line server.

```
tabtp debug tp1 off    (in command mode)
Pactables: debug tp1 off (in "shell" mode)
```

Exit command

This command exits the "shell" mode commands interpreter previously reached by the tabtp -s command.

Help command

This command displays help concerning an on-line server administration command. If no parameter is passed the list of available commands are displayed. If a particular command is specified, its help is displayed on the screen.

EXAMPLE: request to display the syntax of a command start

```
tabtp help start      (in command mode)
Pactables: help start (in "shell" mode)
```

Info command

This command displays the information on the on-line server(s).

This command followed by the server name(info "server_name") gives information from the server configuration files:

```
$TABDIR/assign/tp_serveur/"server_name".cfg  
$TABDIR/assign/tp_serveur/"server_name".ini  
and $TABDIR/assign/"database_name"/"xxxxxx".ini
```

This command alone (info) can give the following information:

- Active (if the server is active),
- Inactive (if the server is inactive),
- Not configured (if the server is declared at \$TABDIR/assign/tp_serveur/tablex.srs level but does not have a configuration file at \$TABDIR/assign/tp_serveur/"server_name".ini) level,
- Error (if the server is stopped abnormally).

Purge command

This command purges a workstation (in other words ends a station's connection). If the interpreter (tabtp) is in command mode, the syntax of the purge command is as follows:

```
"tabtp purge <server_name> <no. of station to be purged>"
```

In "shell" mode there are two syntaxes for the purge command:

- "purge <no. of station to be purged>" if the prompt is different from "Pactables" (the prompt taking the value of the server's name, followed for example by the view command).
- "purge <server_name> <no. of station to be purged>" if the prompt is "Pactables:".

Purge server command

This command purges an on-line server. The syntax for the `purge_server` command is as follows:

```
"tabtp purge_server <server_name>".
```

In "shell" mode the syntax for the command is:

```
"purge_server <server_name>".
```

NOTE: This command deletes the communication files in the TABDIR/commun, TABDIR/tp/save directories and the list of messages (tables IPC) associated with the server.

Shutdown command

This command stops the on-line server. The server's name must be passed as a parameter of the command. A conversations backup file is created in the TABDIR/tp/save directory, and is called the on-line server's name and has extension .ts.

EXAMPLE: stop on-line server tp1

```
pactp shutdown tp1          (in command mode)  
Pactables: shutdown tp1    (in "shell" mode)
```

Start command

This command starts up an on-line server. The server's name must be passed as a parameter of the start command. The initialization file corresponding to the server (on-line server name and extension .ini) must be in the TABDIR/assign/tp_server directory. The server start-up program, `tablaunch`, is launched and starts the `tabserver` process. In case of problems (on-line server blockage) the `tabserver` process can be killed using the `-15 kill` command, or the `-9 kill` followed by destroying the the list of server messages using the `ipcs -q` and `ipcrm -q` commands.

EXAMPLE: start-up the tp1 on-line server

```
tabtp start tp1            (in command mode)  
Pactables: start tp1      (in "shell" mode)
```

Stop command

This command stops an on-line server. The server's name must be passed as a parameter of the stop command. A conversations backup file is created in the TABDIR/tp/save directory and is called the on-line server's name with extension .ts. A confirmation of the 'stop' is requested.

EXAMPLE: stop tp1 on-line server
tabtp stop tp1 (in command mode)
Pactables: stop tp1 (in "shell" mode)

Trace command

This command activates or deactivates a trace on an on-line server. This trace displays the on-line server's response time. The server's name must be passed as a parameter followed by "on" or "off" to activate or deactivate the trace. A file containing the result of the trace is updated and saved in the \$TABDIR/log directory and is called:

"server_name_process_number".LOG

EXAMPLE: start of a trace on tp1 on-line server
tabtp trace tp1 on (in command mode)
Pactables: trace tp1 on (in "shell" mode)

EXAMPLE: stop the trace on tp1 on-line server
tabtp trace tp1 off (in command mode)
Pactables: trace tp1 off (in "shell" mode)

View command

This command displays information concerning an on-line server: the list of connected stations, the user's code, the name of the on-line program (executed or being executed), the time of the start-up of the program and the execution time in milli-seconds.

EXAMPLE: display information on tp1 on-line server

```
tabtp view tp1      (in command mode)
Pactables: view tp1 (in "shell" mode)
```

4.2. TABPARM: PARM TRANSACTION START-UP

PARAMETER TRANSACTION START-UP

The tabparm "database name" command starts the TABPARM transaction on the "database name" database. TABPARM, reserved for the database administrator, is used on the UNIX machine.

TABPARM, the monitor program, is in the TABDIR/bin directory.

Tabparm.ini, the environment variables configuration file, and tabparam, the screen configuration file, are both in the TABDIR/assign/monitors directory.

The other configuration files (xxxxxx.ini) are in the TABDIR/assign/"database name" directory.

END OF TRANSACTION

Exit the transaction: when the user exits the transaction (F6 or CH: FT), the initial grid is displayed; to delete this grid press F12, or an equivalent (refer to paragraph TERMINAL CONFIGURATION).

EXECUTION CONDITIONS

The TABDIR environment variable must be initialized (refer to chapter "INSTALLATION").

The COBPATH environment variable must have access to the "\$TABDIR/tp/gnt" on-line modules.

MONITOR CONFIGURATION

On start-up the monitor executes the tabparm.ini commands file to assign the environment variables it is going to need.

This file is created when the server is created in the TABDIR/assign/monitors directory.

. TABPARM monitor environment variables:

- ROOT : COBOL programs prefix (D8)
and language code
(F = French, A = English)
- FPARAM : screen configuration file

. Database environment variables:

- PAC7TD : table descriptions file
- PAC7TE : error messages file
- PAC7TV : tables contents file
- PAC7TG : user parameters file
- SAVESCR : prompter backup file

TERMINAL CONFIGURATION

Tabparm, the configuration file in the TABDIR/assign/monitors directory, contains the description of the screen and keyboard parameters. This file comprises three parts:

- a list of codes for the display configuration (screen attributes).
- a list of parameters describing the behaviour of the operation of the data entry characters.
- a correspondance table for the terminal controls and the TABPARM functions.

Display parameters

There are eleven display parameters: two color parameters (characters' color and background color) and nine parameters for managing the screen attributes. The two colors available on a monochrome monitor are black (B) and white (W). The first color is the characters' color, the second is the background color.

There are four possible values for the attributes:

- . normal mode N
- . half intensity mode D
- . double intensity mode or bright B
- . reverse mode R

The nine screen attributes fall into the following zones:

- . attribute for normal and protected fields
- . attribute for bright and protected fields
- . attribute for normal fields
- . attribute for bright fields
- . attribute for the current field
- . attribute for the normal underlined fields
- . attribute for the bright underlined fields
- . attribute for the normal flashing fields
- . attribute for the bright flashing fields

Data Entry parameters

There are five parameters for data entry, each can have the value N or Y:

- . the first is not used in this version.
- . the second, if Y, indicates that the cursor will automatically move to the next field when the current field has been filled.
- . the third, if Y, allows for repositioning the cursor in the first field on the screen after the last field.
- . the fourth, if Y, allows for destroying the characters at the end of the current field while in insertion mode.
- . the last is not used in this version.

Keys correspondance table

The last part of the tabparm file describes the correspondances between certain keys and the TABPARM monitor functionalities.

In the current version only the control keys are available. They are noted as follows:

ctrl_A, ctrl_B, ..., ctrl_Z.

The TABPARM functions available are:

. Enter	when the ENTER key is pressed
. Clear	clear the screen
. PF1..PF24	function keys 1 to 24
. Tab	move to the next field
. BackTab	move to the previous field
. NL	new line
. Home	move to the first field
. End	move to the last field
. BeginF	place the cursur at the start of the field
. EndF	place the cursur at the end of the field
. Curs-U	move the cursur to the top
. Curs-D	move the cursur to the bottom
. Curs-L	move the cursur left
. Curs-R	move the cursur right
. BSpace	delete the character before the cursur
. Insert	enter/exit insertion mode
. Delete	delete the character under the cursur
. DeLEOF	delete to the end of the current field
. DelINP	delete the data of the current field
. Recover	recover the screen

By default in configuration files, some controls are already assigned:

ctrl_H	BSpace
ctrl_I	Tab
ctrl_J	NL
ctrl_M	Enter

Other controls, like ctrl-C or ctrl-Z must be used with caution, as they can interfere with the terminal configuration, modifiable by the stty command.

NOTE:

The programming of the function keys is linked with the terminfo configuration file.

Therefore it is advisable to refer to this file for programming the function keys.

4.3. *TABBATCH: BATCH SERVER START-UP*

BATCH MONITOR START-UP

The BATCH monitor, whose executables (tabbatch, tabbatsrv) are in the \$TABDIR/bin directory:

- sets the on-line server(s) to an active or inactive state,
- supplies information concerning the BATCH server(s),
- purges a BATCH server or servers.

To carry out the operations listed above the BATCH monitor commands interpreter (tabbatch) is executed.

There are two possible modes:

- the command mode,
- the "shell" mode.

The command mode consists of typing:

```
tabbatch <command>
```

To pass to "shell" mode type:

```
tabbatch -s.
```

In command mode the BATCH monitor commands can be inserted into a commands file.

Example: The com_table file contains the following commands:

```
# display the server's reports
tabbatch info
# start-up the bat1 BATCH server
tabbatch start bat1
# display the information on bat1 server for ten seconds
tabbatch info bat1
sleep 10
# stop the bat2 on-line server
tabbatch stop bat2
```

The advantage of the "shell" mode is that the interpreter does not have to be restarted for each command; it waits for the next command.

The commands available are as follows:

```
-exit      (exit "shell" mode),  
  
-help      (help on the syntax of a command),  
  
-info      (information on the server(s)),  
  
-purge_server (purge the BATCH server),  
  
-shutdown  (stop the BATCH server without confirmation)  
  
-start     (start the server),  
  
-stop      (stop the BATCH server with confirmation),  
  
-view      (display the server's report).
```

The detailed description of the commands can be found below in the 'Commands' Description' paragraph.

The BATCH modules executed by the server are the same as those of the PRTA procedure (fichiers .gnt in the TABDIR/batch/gnt directory).

To process the print requests for a given database, it is possible to start-up several BATCH servers, which share the same files TLB and TBD.

In the case of several PACTABLE databases, one or more BATCH servers must be defined for each database.

EXECUTION CONDITIONS

The TABDIR environment variable must be initialized (refer to chapter "INSTALLATION").

The COBPATH environment variable must contain the access path to the "\$TABDIR/batch/gnt" on-line modules.

REQUEST RESULTS

When a workstation submits a print job from the LE screen, the following is displayed:

```
JOB SUBMITTED - NUMBER : nnnnn
```

The files created by the PRTA procedure are inserted in the "user code" subdirectory in the "users" directory.

Their code is composed of the request number followed by the type of file generated (refer to the description of the PRTA procedure for more details on this coding).

```
Example: in $TABDIR/users/jean, are      00055.INP
                                           00055.STA
                                           00055.TAB
                                           etc...
```

Only the "useful" files are kept: report of the print execution (.INP), statistics (.STA), table contents (.TAB).

PROCESSING THE RESULTS FILES

The PLBTAGP variable declared in the server configuration file allows the PACAGP commands file to be called after the print generations. Calling this file allows certain tasks to be automated, the nature of which depends on the environments.

An example of the TABAGP file is given on installation in the \$TABDIR/batch/proc directory. This file must be modified to take the specificities of each PACTABLE site into account.

In the example given, the comments explain in particular the parameters that are passed by the batch server to this commands file and can be used for processing the results files.

The TABAGP file is defined by default in the \$TABDIR/batch/proc directory.

BATCH SERVER CONFIGURATION

From it's start-up each server executes a command file of type "server_name".ini to assign environment variables that it will need.

This file is created TABDIR/assign/batch_server directory when the server is created.

. BATCH server environment variables:

- BASENAME : database name
- PAC7LB and PAC7BD: BATCH communicateion files
- PLBTPAR : BATCH assignments file by the BATCH server
- SYSOUT : user's directory,
- PLBTDEL : temporary files delete option,
(YES = destroy, NO = keep)
- PLBTAGP : PACAGP file execution option
(YES = call PACAGP, NO = no call)

. Database files environment variables:

- PAC7TD : table descriptions file
- PAC7TE : error messages file
- PAC7TV : table contents file
- PAC7TG : user parameters file

NOTE:

It is strongly advised to use the tabinstall procedure to create and destroy BATCH servers.

DESCRIPTION OF COMMANDS

Exit command

This command exits the "shell" mode (commands interpreter) previously reached by the tabbatch -s command.

Help command

This command displays help concerning a BATCH server administration command. If no parameter is passed, the list of commands available is displayed. If a particular command is specified, it's help is displayed on the screen.

EXAMPLE: request to display the syntax of the start command

```
tabbatch help start      (in command mode)
Pactables: help start   (in "shell" mode)
```

Info command

This command displays the information on the BATCH server or servers.

This command followed by the server name(info "server_name") gives information from the server configuration files:

```
and      $TABDIR/assign/batch_server/"server_name".ini
         $TABDIR/assign/"database_name"/"xxxxxx".ini
```

This command alone (info) can give the following information:

- Active (if the server is active),
- Inactive (if the server is inactive),
- Not configured (if the server is declared at \$TABDIR/assign/batch_server/tablex.srs level but does not have a configuration file at \$TABDIR/assign/batch_server/"server_name".ini level),
- Error (if the server is stopped abnormally).

Purge_server command

This command purges a BATCH server. The syntax for the purge_server command is as follows:

```
"tabbatch purge_server <server_name>".
```

In "shell" mode the syntax of the command is:

```
"purge_server <server_name>".
```

NOTE: This command deletes the list of messages (tables IPC) associated with the server.

Shutdown command

This command stops the BATCH server without requesting confirmation. The server's name must be passed as a parameter of the command.

EXAMPLE: stop bat1 BATCH server

```
tabbatch shutdown bat1      (in command mode)  
Pactables: shutdown bat1    (in "shell" mode)
```

Start command

This command starts up a BATCH server. The server's name must be passed as a parameter of the start command.

The initialization file corresponding to the server BATCH server's name and extension .ini. must be present in the TABDIR/assign/batch_server directory. The server start-up program, tabbatch, is launched and starts the tabbatsrv process.

In case of problems (BATCH server blockage), the tabbatsrv process can be killed using the -15 kill command, or the -9 kill followed by destroying the list of server messages using the ipcs -q and ipcrm -q commands.

EXAMPLE: start-up the bat1 BATCH server

```
tabbatch start bat1         (in command mode)  
Pactables: start bat1       (in "shell" mode)
```

Stop command

This command stops a BATCH server. The server's name must be passed as a parameter of the stop command.

EXAMPLE: stop the bat1 BATCH server

```
tabbatch stop bat1      (in command mode)
Pactables: stop bat1   (in "shell" mode)
```

View command

This command displays the report for the given BATCH server. For each job submitted to the server, the following information is displayed:

- the line number,
- the job number,
- the PACTABLE user code,
- the workstation number,
- the date and time of job submission,
- the start time of the process,
- the end time of the process,

EXAMPLE: displaying information on the bat1 BATCH server

```
tabbatch view bat1      (in comand mode)
Pactables: view bat1    (in "shell" mode)
```

4.4. PACLINK: UNIX DUMB TERMINAL START-UP

PACLINK: UNIX DUMB WORKSTATION START-UP

The command to start-up an interactive dialog between the station user and the on-line server is:

```
tabwstsock <-trace>
```

The on-line server to which the station will be connected is determined by the station's environment variables:

"PAC_HOST_SERVER_NAME", which contains the name of the UNIX machine where the server is started up.

"PAC_HOST_SERVER_SOCKET", which contains the communication port number used by the server.

In other words, the station will be connected to the server using the "PAC_HOST_SERVER_SOCKET" communication port on the "PAC_HOST_SERVER_NAME" machine.

The "tabwstsock" command is in the TABDIR/bin directory.

"TRACE" OPTION

The -trace option activates the trace.

The daily file supplied by this option is in the form:

wstymmdd.LOG (yy = year, mm = month, dd = day)

It is in the \$TABDIR/log directory.

END OF TRANSACTION

Exit from the transaction: when the user exits the transaction (F6 or CH: FT), the initial grid is displayed.

To delete this grid, press F12, or the equivalent (refer to subchapter "TABPARM: PARM TRANSACTION START-UP", paragraph "TERMINAL CONFIGURATION").

ENVIRONMENT

The environment variable configuration file for "tabwstsock" is called "tabwstsock.ini".

This file is created in the TABDIR/assign/monitors directory at installation.

EXECUTION CONDITIONS

The on-line server must be started up.

CONFIGURATION DESCRIPTION

File: tabwstsock.ini

- FPARAM : terminal configuration file
- PAC7CN : transaction backup file
- TRACE : trace option
(YES = activate, NO = do not activate)
- PAC_HOST_SERVER_NAME : name of UNIX machine where the
on-line server is installed
- PAC_HOST_SERVER_SOCKET : number of on-line server
communication port (socket)

TERMINAL CONFIGURATION

The terminal configuration is managed in the same way as for the "User Parameters" transaction.

Refer to subchapter "TABPARAM: PARM TRANSACTION START-UP".

4.5. PACLINK: WINDOWS DUMB-TERMINAL START-UP

PACLINK: UNIX DUMB WORKSTATION START-UP

On MS-WINDOWS or on WIN-OS/2, the user double clicks on the icon created at installation.

REMINDERS

1. The user must be connected to the local network, to be able to access the Unix machine TCP/IP (with communication by socket).
2. The assignment of the Unix machine and the associated communication port (socket) is done in the GSTCPIP.PRM file.

The user must update this file, in order to adapt the workstation's configuration to that of the on-line server that he wants to connect to.

The communication port associated with the on-line server is created when the on-line server is created on the UNIX machine.

COMMENTS ON THE USE OF WINDOWS DUMB WORKSTATION

The 'dumb terminal' user workstation on WINDOWS uses the GSTCPIP and PACLINK programs.

Refer to the COMMUNICATION DRIVERS and the PACLINK Utility Manual for details on the specifics of this workstation.

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
THE BATCH PROCEDURES

PAGE 57

5

5. THE BATCH PROCEDURES

5.1. INTRODUCTION

BATCH PROCEDURES

The BATCH processes associated with Pactables are regrouped in procedures. The objective of the following chapters is to present each of these procedures and to specify their execution conditions.

For each procedure there is:

- . An introduction comprising:
 - the objective of the procedure,
 - the execution conditions,
 - the actions to carry out in case of abnormalities during the procedure's execution.
- . The description of the user input, the processes and the results obtained, as well as the recommendations for utilization.
- . Description of steps:
 - the notations or parameters used,
 - the list of files used (intermediary and permanents),
 - the return codes for each step.
- . The command procedures.

All the procedures are delivered in the format PROCxxxx, where xxxx represents the procedure code. In the following chapters, the PROC has been omitted as it is understood.

5.2. CLASSIFICATION OF PROCEDURES

PROCEDURES CLASSIFICATION

The procedures associated with BATCH processes are as follows:

- .INTA: Initialization of Pactables files,
- .GETT: Table generation,
- .UPTA: Table updating,
- .PRTA: Table printing,
- .IMTA: Table importation,
- .RETA: Table reorganization,
- .SVTA: Table backup,
- .RSTA: Table restoration,
- .LDTA: List of table descriptions,
- .PMTA: User parameters updating,
- .EXTA: Data extraction,
- .TUTA: Direct consultation of tables,

For the Dispatched Table Management (DTM option):

- .Table description comparison (CDT1, CDT2),
- .Table data comparison (CVTA),

For converting from another platform:

- .Pactables Database migration (TCTA).

REMINDER

Pactables does not include the updating transactions journal.

5.3. STRUCTURE OF PROCEDURE COMMAND FILES

STRUCTURE OF PROCEDURE COMMAND FILES

The BATCH procedures command files are created in the TABDIR/batch/proc directory during installation.

PROCEDURES ADAPTATION TO SITE REQUIREMENTS

The Pactables database manager sometimes has to modify the batch procedures command files.

For example, if the database files are separated onto two disks or if the TE file is moved, the resulting changes in the command files may be considerable.

This is why the Pactables procedures (batch or server start-up) are able to handle all modifications to the standard installation and can minimise the changes to the procedures linked with operational constraints.

The objective of this subchapter is to analyze a batch procedure in order to explain its operation and also to guide the user through its possible adaptations.

STRUCTURE OF A BATCH PROCEDURE

1. The parameters

- General case:

With the exception of CDT1 and CDT2, the BATCH procedures only use one parameter: the database name.

CDT1 and CDT2 need a second parameter: the name of the slave files directory.

- Multi-user procedures case:

Two extra parameters have been added to procedures that are susceptible to being executed simultaneously by several users (CDT1, CDT2, CVTA, EXTA, GETT, IMTA, LDTA, PRTA, UPTA):

- . The first parameter is used as a root for all files in the 'input' and 'tmp' directories in order to differentiate them.
- . The second parameter is used as a suffix or as a sub-directory for the 'input' and 'tmp' directories.

The parameters are valorized or not at the start-up of each BATCH procedure:

- . if the two parameters are blank, the assignment of the files in the 'input' and 'tmp' is not modified,
- . the first parameter may be valorized and the second left blank,
- . the first parameter must contain two characters at most so that the files are visible on DOS or OS/2.
- . the second parameter cannot be valorized without the first,
- . if the second parameter starts with '/' it represents a 'input' and 'tmp' subdirectory, if not it is a suffix of the 'input' and 'tmp' directories; in all cases the creation of all directories is the responsibility of the user.

Both parameters are used in the PACINPUT.ini and PACTMP.ini command files, called in each procedure to initialize the PACINPUT and PACTMP environment variables (see paragraph ASSIGNING AND CODING PARAMETERS).

2. Displaying and validating the parameters

The execution of a procedure begins by the execution of the USAGE.ini command file:

```
. $TABDIR/batch/proc/USAGE.ini
```

This file is created during installation in the \$TABDIR/batch/proc directory.

The USAGE.ini command file controls the procedure and positions the PACRAD and PACSUF environment variables.

These two variable are used in the PACTMP.ini and PACINPUT.ini command files and correspond to root parameters and the multi-users procedures' suffix.

If it detects a fault, the USAGE.ini displays the corresponding error message and stops the procedure with a '20' return code.

If not the execution of the BATCH procedure continues by displaying the directory assignments.

In order to visualize this display, at least during the installation test, the execution must be suspended by a call to the commands file:

```
sh $TABDIR/batch/proc/MSGPAUSE.ini
```

The MSGPAUSE.ini file, created at installation in the TABDIR/batch/proc directory contains:

```
echo ***** Verify your parameters *****  
echo Press Control_C to halt the execution  
echo Press Return to continue  
read REPAUSE
```

If the execution is not to be stopped, the content of MSGPAUSE.ini must be modified by deleting the "read REPAUSE" line.

3. User input content

The procedure commands file contains the description of the user input.

4. Assigning and coding parameters

Each step must be assigned an adequate number of files.

- THE DATABASE FILES

These assignments are carried out by calling the commands file created during installation in the

```
$TABDIR/assign/'database_name' directory.
```

Example of TE file assignment:

```
. $TABDIR/assign/$1/PAC7TE.ini
```

The main interest of these files is to centralize the assignment of each of the database files in a single place. The user who wants to modify the standard location of one file only has to adapt the assignment file.

Note: the same files are used when the servers are started up.

- THE BACKUP FILE

As for the database files, these assignments are carried out via the call of the command files, created during installation in the

```
$TABDIR/assign/"database_name" database.
```

Example of the TE file assignment:

```
. $TABDIR/assign/$1/PACSAVTC.ini
```

By default, the TC backup file is located in the

```
$TABDIR/save/'database_name' directory.
```

The name of the backup file is standardized for all the BATCH procedures using it:

```
input backup (consulted) = TC  
output backup (created by the procedure) = TC.NEW
```

This simplifies the management of files (refer to paragraph "MANAGING THE BACKUP FILE").

- THE TRANSACTION FILES

All the transaction files needed on procedure input are coded MBxxxx (xxxx being the procedure name).

All the transaction files created on procedure output are coded MVxxxx (xxxx being the procedure name). For example the transactions generated by the extraction procedures.

The location of the transaction files is determined by the PACINPUT environment variable, positioned in each procedure, calling the PACINPUT.ini command file:

```
. $TABDIR/assign/$1/PACINPUT.ini
```

The PACINPUT.ini file is created during the installation of PACLAN/X-TABLE and when a database is created in the \$TABDIR/assign/'database_name' directory.

It contains:

Assignment script of the PACINPUT environment variable# ('input' directory)

```
# Parameters description      : $0      = procedure name  
#                             $1      = database name  
#                             $PACRAD = file root  
#                             $PACSUF = directory suffix
```

```
PACINPUT=$TABDIR/input$PACSUF/$1/$PACRAD  
export PACINPUT
```

Example of assignment in the EXTA procedure:

```
PAC7MV=$PACINPUT'MBEXTA'  
export $PAC7MV  
  
PAC7NU=$PACINPUT'MVEXTA'  
export $PAC7NU
```

- THE OUTPUT REPORTS

All the procedure output reports are created in the temporary files directory and their names start with the code of their corresponding procedures. This allows them to be consulted or printed easily. Refer to subchapter "PRINT-OUTS PRODUCED" for details on the coding of these files.

5. End of procedure without error

If no errors are detected, the message, "End of procedure" is displayed.

6. End of procedure with error

If an error is detected in a step, the following steps are not executed.

The name of the program with the error is displayed and also if possible the type of error.

The procedure executes the ERRPAUSE.ini file which stops the procedure and visualizes the error.

The ERRPAUSE.ini file, created on installation, contains:

```
echo "Press Enter to continue"  
read REPAUSE
```

7. Managing the backup file

All procedures creating a backup call a commands file at the end of the procedure (without error).

This file, called TCBACKUP.ini, is in the TABDIR/assign /"database_name" directory.

It is created during installation or when a new database is created and contains (for example with database name d400):

```
# Rotation script of the database backup file
. $TABDIR/assign/d400/PACSAVTC.ini
if -f "$PACSAVTC"
then
    mv -f $TABDIR/save/d400/TC $TABDIR/save/d400/TC-1
fi
mv -f $PACSAVTCNEW $PACSAVTC
mv -f $TABDIR/save/d400/TC.NEW $TABDIR/save/d400/TC
```

Characteristics of TCBACKUP file:

- . use 'mv' to avoid copies being very long,
- . ensure a rotation on the last two versions of the backup files,
- . guarantee that the TC file is the last backup (TC being systematically used as procedure input).

This file does not cover all the operation constraints for all the sites. The database manager will in general adapt it, taking the above characteristics into account.

The TCBACKUP.ini file is used in the SVTA and TCTA procedures.

5.4. *ADVICE ON USE*

ADVICE ON USE

The objective of this subchapter is to make the person in charge of the database aware of the specifics of the PACTABLE procedures executed on UNIX.

Temporary files

For each procedure the user should report to the corresponding chapter for a detailed description of these files.

In all cases, sufficient disk space must be made available on the chosen user directory in order for the procedure to run successfully.

Temporary sort files

When a program executes a sort, the COBOL routines called use a temporary file, independent of those mentioned above.

This file is created by default in the /usr/tmp directory.

Its size may be 3 to 4 times the size of the file to be sorted.

If the default directory is too small the TMPDIR variables allows another directory to be assigned for the temporary sort files:

```
TMPDIR=/tmp2  
export TMPDIR
```

Locating the temporary files

The location of the temporary files is determined by the PACTMP.ini commands file call:

```
. $TABDIR/assign/$1/PACTMP.ini
```

The PACTMP.ini file is created on installing PACTABLE during the creation of a database in the TABDIR/assign/'database_name' directory.

It contains:

```
# Assignment script for temporary files

# Description of parameters : $0      = procedure name
#                          $1      = database name
#                          $PACRAD = file root
#                          $PACSUF = directory suffix

# 'tmp' temporary files directory
PACTMP=$TABDIR/tmp$PACSUF/$1/$PACRAD
export PACTMP

# Sort files directory
# (this directory must be 3 times the size of the file to be
# sorted)
TMPDIR=$TABDIR/tmp/$1
export TMPDIR
```

The PACTMP.ini file initializes the PACTMP environment variable to assign the PACTABLE temporary files.

Example of assignment in the EXTA procedure:

```
PAC7EX=$PACTMP'EX'
export $PAC7EX
```

The PACTMP.ini file initializes the TMPDIR environment variable to assign the sort temporary files directory.

ATTENTION: GENERAL COMMENTS

1. Parameters must be passed to each procedure. All parameters expected when a procedure is called must be present whether they are used or not.
2. When user input is expected in a procedure, even if it is optional, the corresponding transaction file must be present when the procedure is being executed.
3. There is no protection guaranteed in the case where a BATCH procedure updating the PACTABLE database system or evolving files is started up when users are updating the same files interactively. A single person (the database manager) must be able to start up the BATCH procedures updating the PACTABLE databases. The database manager must ensure protection of the database data (by closing the on-line servers for example).
4. The temporary work files created by the BATCH procedures are automatically deleted at the end of the procedure, unless a step does not run properly and sends a return code other than 0.

5.5. PROCEDURES START-UP

START-UP OF BATCH PROCEDURES

The BATCH procedures can only be executed on the UNIX machine used as a Pactables server.

5.6. PRINT-OUTS PRODUCED

PRINT-OUTS PRODUCED

REPORTS CODING

All the reports produced by the BATCH procedures are created in the temporary files directory. These reports are coded with 6 characters plus an extension as follows:

- . the first 4 characters of the report correspond to the BATCH procedure code (EXTA in PROCEXTA),
- . the next 2 characters correspond to the last 2 characters of the file external name (ET in PAC7ET),
- . the extension represents the last 3 characters of it's corresponding program (150 dans PTA150).

Example: EXTA procedure

Program	:	File	-->	Report code on disk
PTA150	:	PAC7ET	-->	EXTAET.150
PTA160	:	PAC7ET	-->	EXTAET.160

5.7. ABNORMAL EXECUTIONS

ABNORMAL EXECUTION

A batch program execution may abend.

For example, input-output errors on the system files or on the Database files cause the interruption of the current program and the display of the following message:

```
PROGR : pppppp   INPUT-OUTPUT ERROR : FILE ff   OP : oo  
STATUS : nn
```

In most cases, examining the status and type of operation allows you to find the cause of the abend.

The tables below indicate standard values for the status and type of operation.

! OO ! OPERATION	! NN ! STATUS
! W ! WRITE	! 21 ! SEQUENCE ERROR
! RW ! REWRITE	! 22 ! DUPLICATE KEY
! RU ! READ UP	! 23 ! NO RECORD FOUND
! OP ! OPEN	! 24 ! BOUNDARY VIOLATION
! CL ! CLOSE	! 30 ! SYSTEM ERROR
! D ! DELETE	! 34 ! BOUNDARY VIOLATION (SEQ.)
! R ! READ	! 35 ! FILE NOT FOUND
! P ! START	! 92 ! LOGICAL ERROR (i.e. OPEN
! RN ! READ NEXT	! ! AN ALREADY OPENED FILE)
! !	! 93 ! LOCKED FILE
! !	! 95 ! INVALID OR INCOMPLETE FILE!
! !	! ! DEFINITION

Some errors, other than input-output errors in a Database file, may also cause the following message to be displayed:

```
Run Time Error nnn
```

where 'nnn' is the error number.

Run Time Error 013 is the most common error. It means that the procedure did not find an input file. In order to find out which file is missing, enter the SET command. This will display the list of allocated files. You can also consult the procedure description in the Batch Procedures Manual. Then, compare this list with the contents of the directories involved.

Most often, it is the input transactions file that is missing (in the 'release\INPUT\db_name' directory : MBxxxx where xxxx is the procedure specific code).

The following subchapter contains the list of the most frequent errors. Each Run Time Error is accompanied by a short explanatory message. If the Run-Time Error is not in the following list, the message is not sufficient and the type of error signals a direct problem in the system programs, contact the VisualAge Pacbase Technical Support and save all listings that could help in analyzing the problem.

ERROR MANAGEMENT IN THE DELIVERED PROCEDURES

At the end of each procedure, if an error has been found, the procedure is interrupted and returns a non-null code. This code can be kept in the \$? variable right after the command for submitting the procedure.

If several procedures are linked, this code prevents the execution of the rest of the procedures.

5.8. LIST OF "RUN TIME ERROR"

LIST OF 'RUN-TIME ERRORS'

This list presents the most common errors and their meaning.

Number	Meaning
-----	-----
004	Invalid file name
005	Invalid device specification
007	No more disk space
009	Directory full or does not exist
013	File not found
026	Block I-O error
027	Device not available
028	Disk space exhausted
033	Physical I-O error
105	Memory allocation error
116	Cannot allocate memory
135	File not found
150	Program abandoned on user request
157	Not enough program memory: object file too large to load
170	System program not found
173	Called program file not found
188	File name too long
198	Not enough program memory: object file too large to load
207	Machine does not exist on the network
208	Network communication error
209	Network communication error
221 !	
222 !>	Error during a sort
223 !	

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
TABLE INITIALIZATION (INTA)

PAGE 76

6

6. TABLE INITIALIZATION (INTA)

6.1. 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).

6.2. 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

6.3. DESCRIPTION OF STEPS

INTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

INITIALIZATION OF FILES: PTAINI

.Input file
PAC7MD

.Output files:
-Table Descriptions File
PAC7TD
-Table Contents File
PAC7TV

.Output report:
-Initialization review
PAC7ED

6.4. EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--      Release xxx Version xxx      --
#@(#)
#@(#)Pactables (R) INTA BATCH Procedure
#@(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                INTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
  echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE INITIALIZATION
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7MD=$PACINPUT'MBINTA'
export PAC7MD
PAC7ED=$PACTMP'INTAED.INI'
export PAC7ED
echo "Execution: PTAINI"
cobrun PTAINI
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTAINI"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
echo "End of procedure"
exit $RETURN
```


VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
TABLE GENERATION (GETT)

PAGE 81

7

7. TABLE GENERATION (GETT)

7.1. INTRODUCTION

GETT: TABLE GENERATION

INTRODUCTION

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

EXECUTION CONDITION

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

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

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

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

USER INPUT

Result of GETD or GETA extraction.

7.2. DESCRIPTION OF STEPS

GETT: DESCRIPTION OF STEPS

UPDATE OF TABLE FILES: PTA250

.Permanent input-output files:

-Table-description file
PAC7TD

-Table-contents file
PAC7TV

.Input transaction file (GETD or GETA output):

-Update transactions
PAC7MD

.Output file

PAC7TK

.Output report:

-Input/output errors on files
PAC7ET

PRINTING OF DESCRIPTIONS: PTA290

.Permanent input file:

-Table-description file
PAC7TD

.Transaction input file:

-Print request
PAC7TE

.Output report:

-Printout of descriptions
PAC7ID

7.3. EXECUTION JCL

```
#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) GETT BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                GETT PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE GENERATION
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7MD=$PACINPUT'MBGETT'
export PAC7MD
PAC7TK=$PACTMP'TK'
export PAC7TK
PAC7ET=$PACTMP'GETTET.250'
export PAC7ET
echo "Execution: PTA250"
cobrun PTA250
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA250"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
PAC7TE=$PACTMP'TK'
export PAC7TE
PAC7ID=$PACTMP'GETTID.290'
export PAC7ID
echo "Execution: PTA290"
cobrun PTA290
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA290"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
```

TABLE GENERATION (GETT)
EXECUTION JCL

PAGE

85

7
3

```
# *****  
echo "End of procedure"  
echo ""  
echo "Deletion of the temporary file TK"  
rm -f $PACTMP'TK'  
exit $RETURN
```

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
TABLE UPDATE (UPTA)

PAGE 86

8

8. TABLE UPDATE (UPTA)

8.1. INTRODUCTION

UPTA: TABLE UPDATING

INTRODUCTION

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

EXECUTION CONDITION

The TV and TD 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.

8.2. USER INPUT

USER INPUT

. One '*'-type line per user:

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

. One 'A'-type line per table to update:

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

8.3. DESCRIPTION OF STEPS

UPTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TABLE UPDATE: PTA300

.Permanent input files:
-Table-description file
 PAC7TD
-Error-message file
 PAC7TE
-User parameters file
 PAC7TG

.Permanent input-output file:
-Table contents file
 PAC7TV

.Input transaction file:
 PAC7MS

.Output file:
-Print requests
 PAC7DE

.Output report:
-Transaction review
 PAC7ET

.Work file:
-Prepared transactions
 PAC7MT

TABLE UPDATE (UPTA)
DESCRIPTION OF STEPS

PAGE

90

8
3

FORMATTING FOR PRINTING: PTA350

.Permanent input files:
-Table-description file
PAC7TD
-Table-contents file
PAC7TV

.Input transaction file:
-Print request
PAC7DE

.Output file:
-Print file
PAC7ET

.Output report:
-Statistics on printing
PAC7EX

PRINTING OF TABLES: PTA360

.Permanent inpput file:
-Table-description file
PAC7TD

.Input Transaction file:
-Print file
PAC7ET

.Output report:
-Printing of tables
PAC7EY

8.4. EXECUTION JCL

```
#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) UPTA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                UPTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE UPDATE
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TG.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7MS=$PACINPUT'MBUPTA'
export PAC7MS
PAC7DE=$PACTMP'DE'
export PAC7DE
PAC7MT=$PACTMP'MT'
export PAC7MT
PAC7ET=$PACTMP'UPTAET.300'
export PAC7ET
echo "Execution: PTA300"
cobrun PTA300
RETURN=$?
case $RETURN in
0)
;;
*)
    echo "Error in executing PTA300"
    echo "Error $RETURN"
    sh $TABDIR/batch/proc/ERRPAUSE.ini
    exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7DE=$PACTMP'DE'
export PAC7DE
PAC7ET=$PACTMP'ET'
export PAC7ET
PAC7EX=$PACTMP'UPTAEX.350'
export PAC7EX
echo "Execution: PTA350"
cobrun PTA350
RETURN=$?
case $RETURN in
0)
;;
```

```
*)
  echo "Error in executing PTA350"
  echo "Error $RETURN"
  sh $TABDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
PAC7ET=$PACTMP'ET'
export PAC7ET
PAC7EY=$PACTMP'UPTAEY.360'
export PAC7EY
echo "Execution: PTA360"
cobrun PTA360
RETURN=$?
case $RETURN in
0)
;;
*)
  echo "Error in executing PTA360"
  echo "Error $RETURN"
  sh $TABDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
;;
esac
# *****
echo "End of procedure"
echo ""
echo "Deletion of the temporary files DE, ET and MT"
rm -f $PACTMP'DE'
rm -f $PACTMP'ET'
rm -f $PACTMP'MT'
exit $RETURN
```

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
TABLE PRINTING (PRTA)

PAGE 93

9

9. TABLE PRINTING (PRTA)

9.1. 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.

9.2. USER INPUT

USER INPUT

.One '*'-type line per user:

! POS.!	! LEN.!	! VALUE	! MEANING
! 2 !	! 1 !	! '*'	! Line code
! 3 !	! 8 !	! uuuuuuu	! 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 !	! DDMCCYY	! Historical account date or
! !	! !	!	! date of the reference description
! !	! !	!	! (if transaction code = 'X')
! 17 !	! 1 !	!	! Sub-schema selection
! !	! !	! blank	! No sub-schema selection
! !	! !	! 1 to 0	! Selected sub-schema number
! 18 !	! 1 !	!	! Sub-system selection
! !	! !	! blank	! No sub-system selection
! !	! !	! 1 to 0	! Selected sub-system number
! 19 !	! 1 !	!	! Print option of the key's data
! !	! !	!	! elements
! !	! !	! blank	! Printing of concatenated data
! !	! !	!	! elements
! !	! !	! 'O'	! Printing of separated data
! !	! !	!	! elements

9.3. DESCRIPTION OF STEPS

PRTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

EXTRACTION OF TABLES FOR PRINTING: PTA320

.Permanent input files:
-Table-description File
PAC7TD
-Error-message file
PAC7TE
-Table-contents File
PAC7TV
-User-parameter file
PAC7TG

.Input transaction file:
-Update transactions
PAC7CA

.Output file:
-Print requests
PAC7DE

.Output report:
-Transaction review
PAC7XE

TABLE PRINTING (PRTA)
DESCRIPTION OF STEPS

PAGE

97

9
3

PREPARATION FOR PRINTING: PTA350

.Permanent input files:
-Table-description File
PAC7TD
-Table-contents file
PAC7TV

.Input transaction file:
-Print requests
PAC7DE

.Output file:
-Print file
PAC7ET

.Output report:
-Statistics on printing
PAC7EX

PRINTING OF TABLES: PTA360

.Permanent input file:
-Tables-description file
PAC7TD

.Input transaction file:
-Print file
PAC7ET

.Output report:
-Printing of tables

9.4. EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--      Release xxx Version xxx      --
#@(#)
#@(#)Pactables (R) PRTA BATCH Procedure
#@(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                PRTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
  echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE PRINTING
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TG.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7CA=$PACINPUT'MBPRTA'
export PAC7CA
PAC7DE=$PACTMP'DE'
export PAC7DE
PAC7XE=$PACTMP'PRTAXE.320'
export PAC7XE
echo "Execution: PTA320"
cobrun PTA320
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA320"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7DE=$PACTMP'DE'
export PAC7DE
PAC7ET=$PACTMP'ET'
export PAC7ET
PAC7EX=$PACTMP'PRTAEX.350'
export PAC7EX
echo "Execution: PTA350"
cobrun PTA350
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA350"
```

```
echo "Error $RETURN"  
sh $TABDIR/batch/proc/ERRPAUSE.ini  
exit $RETURN  
;;  
esac  
# *****  
. $TABDIR/assign/$1/PAC7TD.ini  
PAC7ET=$PACTMP'ET'  
export PAC7ET  
PAC7EY=$PACTMP'PRTAEY.360'  
export PAC7EY  
echo "Execution: PTA360"  
cobrun PTA360  
RETURN=$?  
case $RETURN in  
0)  
;;  
*)  
echo "Error in executing PTA360"  
echo "Error $RETURN"  
sh $TABDIR/batch/proc/ERRPAUSE.ini  
exit $RETURN  
;;  
esac  
# *****  
echo "End of procedure"  
echo ""  
echo "Deletion of the temporary files DE and ET"  
rm -f $PACTMP'DE'  
rm -f $PACTMP'ET'  
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	100
TABLES - EXPLOITATION & INSTALLATION		
TABLE IMPORT (IMTA)		10

10. TABLE IMPORT (IMTA)

10.1. INTRODUCTION

TABLE IMPORT (IMTA): INTRODUCTION

This procedure imports external tables into the existing Pactables files.

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

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

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

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

EXECUTION CONDITION

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

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

If the table to be imported is large, 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 2.0 and higher releases.

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

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

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

10.2. USER INPUT

USER INPUT

.One '*'-type line per user:

```
+-----+-----+-----+-----+
! POS.! LEN.! VALUE      ! MEANING      !
+-----+-----+-----+-----+
!  2 !  1 ! '*'          ! Line code    !
!  3 !  8 ! uuuuuuuu    ! User code    !
! 11 !  8 ! pppppppp    ! Password     !
+-----+-----+-----+-----+
```

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

```
+-----+-----+-----+-----+
! POS.! LEN.! VALUE      ! MEANING      !
+-----+-----+-----+-----+
!  2 !  1 ! 'A'         ! Line code    !
!  3 !  6 ! tttttt     ! Number of the table to be imported!
!  9 !  8 ! DDMMCCYY   ! Table date (optional) !
+-----+-----+-----+-----+
```

10.3. DESCRIPTION OF STEPS

IMTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TABLE CHECK AND UPDATE: PTA310

.Permanent input files:
-Table-description file
PAC7TD
-Error-message file
PAC7TE
-User-parameter file
PAC7TG

.Permanent input-output file:
-Table-contents files
PAC7TV

.Input file:
-Request transactions
PAC7MV
-External table-file
PAC7NK

.Output file:
-Print requests
PAC7DE

.Output report:
-Execution report
PAC7ET

TABLE IMPORT (IMTA)
DESCRIPTION OF STEPS

PAGE

105

10
3

FORMATTING OF PRINTOUT: PTA350

.Permanent input files:
-Table-descriptions file
PAC7TD
-Table-contents file
PAC7TV

.Input transaction file:
-Print requests
PAC7DE

.Output file:
-Print file
PAC7ET

.Output report:
-Printing statistics
PAC7EX

PRINTING: PTA360

.Permanent input file:
-Table-description file
PAC7TD

.Input transaction file:
-Print file
PAC7ET

.Output report:
-Table printout
PAC7EY

10.4. EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--          Release xxx Version xxx          --
#@(#)
#@(#)Pactables (R) IMTA BATCH Procedure
#@(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                IMTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE IMPORT
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TG.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7MV=$PACINPUT'MBIMTA'
export PAC7MV
PAC7NK=$PACINPUT'MTIMTA'
export PAC7NK
PAC7DE=$PACTMP'DE'
export PAC7DE
PAC7ET=$PACTMP'IMTAET.310'
export PAC7ET
echo "Execution: PTA310"
cobrun PTA310
RETURN=$?
case $RETURN in
0)
;;
*)
    echo "Error in executing PTA310"
    echo "Error $RETURN"
    sh $TABDIR/batch/proc/ERRPAUSE.ini
    exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TV.ini
PAC7DE=$PACTMP'DE'
export PAC7DE
PAC7ET=$PACTMP'ET'
export PAC7ET
PAC7EX=$PACTMP'IMTAEX.350'
export PAC7EX
echo "Execution: PTA350"
cobrun PTA350
RETURN=$?
case $RETURN in
0)
;;
;;
```

```
*)  
  echo "Error in executing PTA350"  
  echo "Error $RETURN"  
  sh $TABDIR/batch/proc/ERRPAUSE.ini  
  exit $RETURN  
  ; ;  
esac  
# *****  
  . $TABDIR/assign/$1/PAC7TD.ini  
  PAC7ET=$PACTMP'ET'  
  export PAC7ET  
  PAC7EY=$PACTMP'IMTAEY.360'  
  export PAC7EY  
  echo "Execution: PTA360"  
  cobrun PTA360  
  RETURN=$?  
  case $RETURN in  
  0)  
    ; ;  
  *)  
    echo "Error in executing PTA360"  
    echo "Error $RETURN"  
    sh $TABDIR/batch/proc/ERRPAUSE.ini  
    exit $RETURN  
    ; ;  
  esac  
# *****  
  echo "End of procedure"  
  echo ""  
  echo "Deletion of the temporary files DE and ET"  
  rm -f $PACTMP'DE'  
  rm -f $PACTMP'ET'  
  exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	108
TABLES - EXPLOITATION & INSTALLATION		
TABLE REORGANIZATION (RETA)		11

11. TABLE REORGANIZATION (RETA)

11.1. INTRODUCTION

RETA: TABLE REORGANIZATION

INTRODUCTION

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

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

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

EXECUTION CONDITION

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

11.2. USER INPUT

USER INPUT

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

! POS.!	! LEN.!	! VALUE	! MEANING
! 2 !	! 1 !	! '*'	! Line code
! 3 !	! 8 !	! '*****'	! Table Manager code
! 11 !	! 8 !	! pppppppp	! Table Manager password

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

! POS.!	! LEN.!	! VALUE	! MEANING
! 1 !	! 1 !		! Action code
! !	! !	! 'S'	! Historical account to 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.

11.3. DESCRIPTION OF STEPS

RETA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

REORGANIZATION OF TABLE CONTENTS: PTA400

.Permanent input files:
-Table-description file
PAC7TD
-Error-message file
PAC7TE
-Table-contents file
PAC7TV
-User parameter file
PAC7TG

.Input transaction file:
-Reorganization requests
PAC7DR

.Output file:
-Reorganized-contents file
PAC7TX
-Reorganized-table list file
PAC7DE

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

.Output report:
-Transaction report
PAC7IR

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

VALIDIDATION OF TABLE CONTENTS: PTA410

.Updating input file :
-Reorganization requests
PAC7MB

.Input file:
-Reorganized-contents file
PAC7TX

.Output file:
-Validated-contents file
PAC7TW

REORGANIZATION OF TABLE-DESCRIPTIONS: PTA420

.Permanent input file:
-Table-description file
PAC7TD

.Input file:

TABLE REORGANIZATION (RETA)
DESCRIPTION OF STEPS

PAGE

112

11
3

-Reorganized-table list file
PAC7DE

.Output files:
-Reorganized table-description file
PAC7TS
-Table-description print request
PAC7ML

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

BUILDING OF BACKUP FILE: PTA430

.Input files:
-Validated-contents file
PAC7TW
-Reorganized-description file
PAC7TS

.Output file:
-Backup file resulting from
reorganization
PAC7TC

TG FILE BACKUP: PTASVG

.Permanent input file:
-User-parameter file
PAC7TG

.Output file:
-Table backup
PAC7TC

11.4. EXECUTION JCL

```

#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) RETA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                      RETA PROCEDURE"
echo "                      ====="
echo "Directory 'assign'          : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'           : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'         : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
  echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE REORGANIZATION
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TV.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7DR=$PACINPUT'MBRETA'
export PAC7DR
PAC7DE=$PACINPUT'MVRETA'
export PAC7DE
PAC7TX=$PACTMP'TX'
export PAC7TX
PAC7IR=$PACTMP'RETAIR.400'
export PAC7IR
echo "Execution: PTA400"
cobrun PTA400
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA400"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
PAC7TX=$PACTMP'TX'
export PAC7TX
PAC7TW=$PACTMP'TW'
export PAC7TW
PAC7MB=$PACINPUT'MBRETA'
export PAC7MB
echo "Execution: PTA410"
cobrun PTA410
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA410"

```

```
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
PAC7TS=$PACTMP'TS'
export PAC7TS
PAC7DE=$PACINPUT'MVRETA'
export PAC7DE
PAC7ML=$PACINPUT'MWRETA'
export PAC7ML
echo "Execution: PTA420"
cobrun PTA420
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA420"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
PAC7TW=$PACTMP'TW'
export PAC7TW
PAC7TS=$PACTMP'TS'
export PAC7TS
echo "Execution: PTA430"
cobrun PTA430
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA430"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TG.ini
. $TABDIR/assign/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
echo "Execution: PTASVG"
cobrun PTASVG
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTASVG"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
echo "End of procedure"
echo ""
echo "Calling the file TCBACKUP.CMD"
. $TABDIR/assign/$1/TCBACKUP.ini
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	115
TABLES - EXPLOITATION & INSTALLATION		
BACKUP (SVTA)		12

12. BACKUP (SVTA)

12.1. 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.

12.2. DESCRIPTION OF STEPS

SVTA: DESCRIPTION OF STEPS

TD BACKUP: PTASVD

.Permanent input files:
-Table-description file
PAC7TD
.Output file:
-Table backup
PAC7TC

TV BACKUP: PTASVV

.Permanent input file:
-Table-contents file
PAC7TV
.Output file:
-Table backup
PAC7TC

TG BACKUP: PTASVG

.Permanent input file:
-User-parameter file
PAC7TG
.Output file:
-Table backup
PAC7TC

12.3. EXECUTION JCL

```
#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) SVTA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                               SVTA PROCEDURE"
echo "                               ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE BACKUP
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
echo "Execution: PTASVD"
cobrun PTASVD
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTASVD"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TV.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
echo "Execution: PTASVV"
cobrun PTASVV
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTASVV"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TG.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
echo "Execution: PTASVG"
cobrun PTASVG
```

```
RETURN=$?  
case $RETURN in  
0)  
  ;;  
*)  
  echo "Error in executing PTASVG"  
  echo "Error $RETURN"  
  sh $TABDIR/batch/proc/ERRPAUSE.ini  
  exit $RETURN  
  ;;  
esac  
# *****  
echo "End of procedure"  
echo ""  
echo "Calling the file TCBACKUP.ini"  
. $TABDIR/assign/$1/TCBACKUP.ini  
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	120
TABLES - EXPLOITATION & INSTALLATION		
PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)		13

13. PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)

13.1. INTRODUCTION

TRANSFER OF A DATABASE FROM ANOTHER PLATFORM

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

The Database backup is sorted according to the format 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.

NOTE:

1. BACKUP TRANSFER

The transfer of the TC backup from the original site to the local network where Pactables is to be installed, is handled by the user. The parameters to be specified during the transfer are: the data files (DATA), which must be converted to the ASCII format and must contain the control characters for the recordings ends (CRLF in general, or LINE_FEED).

2. DISK SPACE

The TCTA procedure, described below, consists in sorting the TC backup according to an ASCII sequence. The sort is carried out in 3 different programs to minimize the required disk space. However, the procedure needs 3 to 4 times the equivalent of the original file.

This procedure does not include the deletion of the TC.INI file. It is possible to delete this file at the end of the first step, if the user needs more disk space.

For execution reasons, it is recommended to create the work and sort files on the Pactables server disk which executes the programs.

13.2. DESCRIPTION OF STEPS

TCTA: DESCRIPTION OF STEPS

TC BACKUP SPLIT: PTATC1

.Input backup file
PAC7TC

.Output work files:
-Table-description sequential image
PAC7SD
-Table-contents sequential image
PAC7SV
-Parameter sequential image
PAC7SG

TABLE-DESCRIPTION SORT: PTATCD

.Input work file:
-Table-description sequential image
PAC7SD

.Output work file:
-Sorted table descriptions
PAC7AD

TABLE-CONTENTS SORT: PTATCV

.Input work file:
-Sequential image of table-contents
PAC7SV

.Output work file:
-Sorted table-contents
PAC7AV

USER-PARAMETER SORT: PTATCG

.Input work file:
Sequential image of parameters
PAC7SG

.Output work file:
-Sorted user parameters
PAC7AG

RECONSTITUTION OF THE TC BACKUP: PTATC2

.Permanent output file:
-TC backup in EBCDIC format
PAC7TC

.Input work files:
-Table-description sequential image
PAC7AD
-Sequential image of contents
PAC7AV
-Sequential image of parameters
PAC7AG

13.3. EXECUTION JCL

```
#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) TCTA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                      TCTA PROCEDURE"
echo "                      ====="
echo "Directory 'assign'          : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'           : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'         : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo ""
. $TABDIR/assign/$1/PACSAVTC.ini
echo " Note:"
echo "  Input TC file (from mainframe) : $PACSAVTC.INI"
echo "  Output TC file (ASCII format)  : $PACSAVTC"
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * PactablesS : DATABASE RETRIEVAL FOR MIGRATION
# *****
PAC7TC=$PACSAVTC.INI
export PAC7TC
PAC7SD=$PACTMP'SD'
export PAC7SD
PAC7SG=$PACTMP'SG'
export PAC7SG
PAC7SV=$PACTMP'SV'
export PAC7SV
echo "Execution: PTATC1"
cobrun PTATC1
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTATC1"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
PAC7SD=$PACTMP'SD'
export PAC7SD
PAC7AD=$PACTMP'AD'
export PAC7AD
echo "Execution: PTATCD"
cobrun PTATCD
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTATCD"
echo "Error $RETURN"
```

PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)
EXECUTION JCL

13
3

```

sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
echo "Deletion of the temporary file SD"
rm -f $PACTMP'SD'
# *****
PAC7SV=$PACTMP'SV'
export PAC7SV
PAC7AV=$PACTMP'AV'
export PAC7AV
echo "Execution: PTATCV"
cobrun PTATCV
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTATCV"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
echo "Deletion of the temporary file SV"
rm -f $PACTMP'SV'
# *****
PAC7SG=$PACTMP'SG'
export PAC7SG
PAC7AG=$PACTMP'AG'
export PAC7AG
echo "Execution: PTATCG"
cobrun PTATCG
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTATCG"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
echo "Deletion of the temporary file SG"
rm -f $PACTMP'SG'
# *****
PAC7TC=$PACSAVTCNEW
export PAC7TC
PAC7AD=$PACTMP'AD'
export PAC7AD
PAC7AG=$PACTMP'AG'
export PAC7AG
PAC7AV=$PACTMP'AV'
export PAC7AV
echo "Execution: PTATC2"
cobrun PTATC2
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTATC2"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
echo "Deletion of the temporary files AD, AG and AV"
rm -f $PACTMP'AD'
rm -f $PACTMP'AG'
rm -f $PACTMP'AV'
# *****

```

PACTABLES TRANSFER FROM ANOTHER PLATFORM (TCTA)
EXECUTION JCL

PAGE

126

13
3

```
echo "End of procedure"  
echo "  
echo "Calling the file TCBACKUP.ini"  
. $TABDIR/assign/$1/TCBACKUP.ini  
exit $RETURN
```

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
RESTORATION (RSTA)

PAGE 127

14

14. RESTORATION (RSTA)

14.1. 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.

14.2. DESCRIPTION OF STEPS

RSTA: DESCRIPTION OF STEPS

RESTORATION OF TD: PTARSD

.Permanent output file:
-Table-description file
PAC7TD

.Permanent input file:
-Table backup
PAC7TC

RESTORATION OF TV: PTARSV

.Permanent output file:
-Table-contents file
PAC7TV

.Permanent input file:
-Table backup
PAC7TC

RESTORATION OF TG: PTARSG

.Permanent output file:
-User parameter file
PAC7TG

.Permanent input file:
-Table backup
PAC7TC

14.3. EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--          Release xxx Version xxx          --
#@(#)
#@(#)Pactables (R) RSTA BATCH Procedure
#@(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                RSTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE RESTORATION
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PACSAVTC.ini
PAC7TC=$PACSAVTC
export PAC7TC
echo "Execution: PTARSD"
cobrun PTARSD
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTARSD"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TV.ini
PAC7TC=$PACSAVTC
export PAC7TC
echo "Execution: PTARSV"
cobrun PTARSV
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTARSV"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TG.ini
PAC7TC=$PACSAVTC
export PAC7TC
echo "Execution: PTARSG"
cobrun PTARSG
```

RESTORATION (RSTA)
EXECUTION JCL

PAGE

131

14
3

```
RETURN=$?  
case $RETURN in  
0)  
  ;;  
*)  
  echo "Error in executing PTARSG"  
  echo "Error $RETURN"  
  sh $TABDIR/batch/proc/ERRPAUSE.ini  
  exit $RETURN  
  ;;  
esac  
# *****  
echo "End of procedure"  
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	132
TABLES - EXPLOITATION & INSTALLATION		
LIST OF TABLE DESCRIPTIONS (LDTA)		15

15. LIST OF TABLE DESCRIPTIONS (LDTA)

15.1. 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.

15.2. USER INPUT

USER INPUT

.A 'Z'-type line per print request:

! POS.!	LEN.!	VALUE	! MEANING	!
! 2 !	1 !	'Z'	! Line code	!
! 5 !	4 !		! Print request	!
! !	!	'TLS '	! List of table descriptions	!
! !	!	'TDS '	! Table description	!
! 9 !	6 !	tttttt	! Table number	!
! 23 !	8 !	MMDDCCYY	! Historical account date	!

NOTE:

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

15.3. DESCRIPTION OF STEPS

LDTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TABLE-DESCRIPTION PRINTING: PTA290

.Permanent input file:
-Table-description file
PAC7TD

.Input transaction file:
-Print request
PAC7TE

.Output report:
-Table-description printout
PAC7ID

15.4. EXECUTION JCL

```

#!/bin/sh
#(#)
#(#)--          Release xxx Version xxx          --
#(#)
#(#)Pactables (R) LDTA BATCH Procedure
#(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                LDTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
  echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : LIST OF TABLE DESCRIPTIONS
# *****
. $TABDIR/assign/$1/PAC7TD.ini
PAC7TE=$PACINPUT'MBLDTA'
export PAC7TE
PAC7ID=$PACTMP'LDTAID.290'
export PAC7ID
echo "Execution: PTA290"
cobrun PTA290
RETURN=$?
case $RETURN in
0)
  ;;
*)
  echo "Error in executing PTA290"
  echo "Error $RETURN"
  sh $TABDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
  ;;
esac
# *****
echo "End of procedure"
exit $RETURN

```


VisualAge Pacbase - Operations Manual	PAGE	137
TABLES - EXPLOITATION & INSTALLATION		
PARAMETER UPDATE (PMTA)		16

16. PARAMETER UPDATE (PMTA)

16.1. 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.

16.2. USER INPUT

USER INPUT

'TA'-line: user parameter updating:

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

'TC'-line: access authorizations per table:

!POS.!	!LEN.!	! VALUE	!MEANING	!
! 1 !	! 1 !	!	!Action code	!
!	!	! blank	!Creation or modification	!
!	!	! 'C'	!Creation	!
!	!	! 'M'	!Modification	!
!	!	! 'D'	!Deletion	!
! 2 !	! 8 !	!uuuuuuuu!	!User code	!
! 10 !	! 2 !	! 'TC'	!Line code	!
! 12 !	! 6 !	! tttttt	!Table code	!
! 18 !	! 3 !	! nnn	!Line number	!
! 21 !	! 60 !	!	!Access authorizations: 20 access	!
!	!	!	!authorizations may be entered in this	!
!	!	!	!field, with, for each authorization:	!
!	! 1 !	! n	! Sub-schema number	!
!	! 1 !	! n	! Sub-system number	!
!	! 1 !	! x	! Authorization (0,1 or 2)	!
!	!	!	! ('*' for all sub-schemas and	!
!	!	!	! sub-systems)	!

'TJ'-line: control cards:

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

NOTE:

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

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

16.3. DESCRIPTION OF STEPS

PMTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

PARAMETER UPDATE: PTA100

.Permanent input files:
-Table-description file
 PAC7TD
-Error-message file
 PAC7TE

.Permanent input-output file:
-User parameter file
 PAC7TG

.Input transaction file:
-Extraction requests
 PAC7MV

.Output file:
-Parameter printing requests
 PAC7NU

.Output report:
-Printing of descriptions
 PAC7ET

PRINTING OF USER PARAMETERS: PTA120

.Permanent input files:
-Table description file
 PAC7TD
-User parameter file
 PAC7TG

.Input transaction file:
-Print requests
 PAC7NU

.Output report:
-Printing of user parameters
 PAC7ET

16.4. EXECUTION JCL

```
#!/bin/sh
#(#)
#(#)--          Release xxx Version xxx          --
#(#)
#(#)Pactables (R) PMTA BATCH Procedure
#(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                PMTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : USER PARAMETER UPDATING
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7MV=$PACINPUT'MBPMTA'
export PAC7MV
PAC7NU=$PACTMP'NU'
export PAC7NU
PAC7ET=$PACTMP'PMTAET.100'
export PAC7ET
echo "Execution: PTA100"
cobrun PTA100
RETURN=?
case $RETURN in
0)
; ;
*)
echo "Error in executing PTA100"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
; ;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7NU=$PACTMP'NU'
export PAC7NU
PAC7ET=$PACTMP'PMTAET.120'
export PAC7ET
echo "Execution: PTA120"
cobrun PTA120
RETURN=?
case $RETURN in
0)
; ;
*)
echo "Error in executing PTA120"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
```

PARAMETER UPDATE (PMTA)
EXECUTION JCL

PAGE

143

16
4

```
;;  
esac  
# *****  
echo "End of procedure"  
echo ""  
echo "Deletion of the temporary file NU"  
rm -f $PACTMP'NU'  
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	144
TABLES - EXPLOITATION & INSTALLATION		
TABLE EXTRACTION (EXTA)		17

17. TABLE EXTRACTION (EXTA)

17.1. INTRODUCTION

EXTA: TABLE EXTRACTION

INTRODUCTION

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

EXECUTION CONDITION

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

17.2. USER INPUT

USER INPUT

.One '*'-type line per user:

```
+-----+-----+-----+-----+
!POS.!LEN.! VALUE !MEANING !
+-----+-----+-----+-----+
! 2 ! 1 ! '*' !Line code !
! 3 ! 8 ! !uuuuuuu!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 ! '/' !
+-----+-----+-----+-----+
```

17.3. DESCRIPTION OF STEPS

EXTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

EXTRACTION OF TABLE DATA: PTA150

.Permanent input files:
-Table-description file
PAC7TD
-Error message file
PAC7TE
-Table contents file
PAC7TV
-User parameter file
PAC7TG

.Input transaction file:
-Extraction requests
PAC7MV

.Output file:
-Extracted transactions
PAC7EX

.Output report:
-Transaction review
PAC7ET

PRINTING OF EXTRACTED TRANSACTIONS: PTA160

.Permanent input file:
-Table description file
PAC7TD

.Input transaction file:
-Extracted transactions
PAC7EX

.Output report:
-Printing of extracted data
PAC7ET

.Output file:
-Extracted transactions
PAC7NU

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

17.4. EXECUTION JCL

```

#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) EXTA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                EXTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
  echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE EXTRACTION
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TV.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7MV=$PACINPUT'MBEXTA'
export PAC7MV
PAC7EX=$PACTMP'EX'
export PAC7EX
PAC7ET=$PACTMP'EXTAET.150'
export PAC7ET
echo "Execution: PTA150"
cobrun PTA150
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA150"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
PAC7EX=$PACTMP'EX'
export PAC7EX
PAC7NU=$PACINPUT'MVEXTA'
export PAC7NU
PAC7ET=$PACTMP'EXTAET.160'
export PAC7ET
echo "Execution: PTA160"
cobrun PTA160
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTA160"
echo "Error $RETURN"

```

```
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
echo "End of procedure"
echo ""
echo "Deletion of the temporary file EX"
rm -f $PACTMP'EX'
exit $RETURN
```

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
DIRECT CONSULTATION OF TABLES (TUTA)

PAGE 150

18

18. DIRECT CONSULTATION OF TABLES (TUTA)

18.1. INTRODUCTION

TUTA: DIRECT CONSULTATION OF TABLES

INTRODUCTION

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

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

EXECUTION CONDITION

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

The TUTA procedure defines both files in the second step.

18.2. USER INPUT

USER INPUT

.One '*'-type line :

```
+-----+-----+-----+-----+
!POS.!LEN.! VALUE !MEANING !
+-----+-----+-----+-----+
! 2 ! 1 ! '*' !Line code !
! 3 ! 8 ! !uuuuuuu!User code !
! 11 ! 8 ! !pppppppp!Password !
+-----+-----+-----+-----+
```

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

```
+-----+-----+-----+-----+
!POS.!LEN.! VALUE !MEANING !
+-----+-----+-----+-----+
! 2 ! 1 ! 'A' !Line code !
! 3 ! 6 ! !tttttt !Table number !
! 9 ! 8 ! !DDMMCCYY!Historical account date !
+-----+-----+-----+-----+
```

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

18.3. DESCRIPTION OF STEPS

TUTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

DIRECT CONSULTATION OF TABLES: PTAU80

.Permanent input files:
-Table-description file
 PAC7TD
-Error-message file
 PAC7TE
-Table-contents file
 PAC7TV
-User-parameter file
 PAC7TG

.Input transaction file:
-Request transactions
 PAC7MX

.Permanent output files:
-Table-description file
 PAC7AD
-Table-contents file
 PAC7AV

.Output report:
-Transaction report

18.4. EXECUTION JCL

```
#!/bin/sh
#(#)
#(#)--          Release xxx Version xxx          --
#(#)
#(#)Pactables (R) TUTA BATCH Procedure
#(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                TUTA PROCEDURE"
echo "                                ====="
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : TABLE PRODUCTION TURNOVER
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TV.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7MX=$PACINPUT'MBTUTA'
export PAC7MX
. $TABDIR/assign/$1/PAC7AD.ini
. $TABDIR/assign/$1/PAC7AV.ini
PAC7ET=$PACTMP'TUTAET.U80'
export PAC7ET
echo "Execution: PTAU80"
cobrun PTAU80
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTAU80"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
echo "End of procedure"
exit $RETURN
```

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
DISPATCHED TABLE MANAGEMENT (DTM OPTION)

PAGE 155

19

19. DISPATCHED TABLE MANAGEMENT (DTM OPTION)

19.1. TABLE DESCRIPTION COMPARISON (CDT1-CDT2)

DISPATCHED TABLE MANAGER (DTM)

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

TABLE DESCRIPTION COMPARISON

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

This file may be used as input in the CDT2 procedure in order to update the 'outdated' version of the table description.

EXECUTION CONDITION

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

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

19.2. USER INPUT (CDT1)

USER INPUT

.One '*' -type line per user:

!POS.!	!LEN.!	! VALUE	!MEANING	!
! 2 !	! 1 !	! '*'	!Line code	!
! 3 !	! 8 !	!uuuuuuuu!	User code	!
! 11 !	! 8 !	!pppppppp!	Password	!

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

!POS.!	!LEN.!	! VALUE	!MEANING	!
! 2 !	! 1 !	! 'A'	!Line code	!
! 3 !	! 6 !	! tttttt	!Table number	!

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

19.3. DESCRIPTION OF STEPS (CDT1)

CDT1: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

CHECK OF TRANSACTIONS: PTAD05

.Permanent input files:
- 'Master' table-description file
PAC7TD
- Error-message file
PAC7TE
- User-parameter file
PAC7TG

.Input transaction file:
- Comparison request transactions
PAC7MV

.Output file:
- Validated comparison request transactions
PAC7MX

.Output report:
- Transaction report
PAC7ET

TABLE-DESCRIPTION COMPARISON AND EXTRACTION: PTAD10

.Permanent input files:
- 'Master' table-description file
PAC7TD
- Error message file
PAC7TE
- 'Slave' table-description file
PAC7TS

.Input transaction file:
- Validated transactions
PAC7MX

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

.Output report:
- Extraction printout
PAC7ET

19.4. EXECUTION JCL (CDT1)

```

#!/bin/sh
#@(#)
#@(#)--          Release xxx Version xxx          --
#@(#)
#@(#)Pactables (R) CDT1 BATCH Procedure
#@(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                CDT1 PROCEDURE"
echo "                                ====="
echo "  Please note the specific parameters"
echo "  -----"
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$3" ]
then
    echo "Radical 'tmp' and 'input' files : $3"
fi
echo "Slave files directory            : $2"
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : DISPATCHED TABLE MANAGEMENT
# *                TABLE DESCRIPTION COMPARISON
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7MV=$PACINPUT'MBCDT1'
export PAC7MV
PAC7MX=$PACTMP'MX'
export PAC7MX
PAC7ET=$PACTMP'CDT1ET.D05'
export PAC7ET
echo "Execution: PTAD05"
cobrun PTAD05
RETURN=$?
case $RETURN in
0)
; ;
*)
echo "Error in executing PTAD05"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
; ;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
PAC7TS=$2/TD
export PAC7TS
PAC7MX=$PACTMP'MX'
export PAC7MX
PAC7TX=$PACINPUT'MBCDT2'
export PAC7TX
PAC7ET=$PACTMP'CDT1ET.D10'
export PAC7ET
echo "Execution: PTAD10"
cobrun PTAD10
RETURN=$?

```

```
case $RETURN in
0)
;;
*)
echo "Error in executing PTAD10"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
echo "End of procedure"
echo ""
echo "Deletion of the temporary file MX"
rm -f $PACTMP'MX'
exit $RETURN
```


19.5. 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
- Error-message file
PAC7TE

.Output file:
- File of table-contents associated to the
'slave' table-description file
PAC7TV

.Input transaction file:
- Result extracted from comparison in the
CDT1 procedure
PAC7TX

.Output report:
- Update report
PAC7ET

19.6. EXECUTION JCL (CDT2)

```
#!/bin/sh
#@(#)
#@(#)--          Release xxx Version xxx          --
#@(#)
#@(#)Pactables (R) CDT2 BATCH Procedure
#@(#)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                                CDT2 PROCEDURE"
echo "                                ====="
echo "  Please note the specific parameters"
echo "  -----"
echo "Directory 'assign'                : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$3" ]
then
  echo "Radical 'tmp' and 'input' files : $3"
fi
echo "Slave files directory            : $2"
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : DISPATCHED TABLE MANAGEMENT
# *                TABLE DESCRIPTION UPDATE
# *****
. $TABDIR/assign/$1/PAC7TE.ini
PAC7TD=$2/TD
export PAC7TD
PAC7TV=$2/TV
export PAC7TV
PAC7TX=$PACINPUT'MBCDT2'
export PAC7TX
PAC7ET=$PACTMP'CDT2ET.D20'
export PAC7ET
echo "Execution: PTAD20"
cobrun PTAD20
RETURN=$?
case $RETURN in
0)
;;
*)
echo "Error in executing PTAD20"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
;;
esac
# *****
echo "End of procedure"
exit $RETURN
```

19.7. TABLE CONTENTS UPDATE (CVTA)

CVTA: COMPARISON AND UPDATING OF TABLE CONTENTS

INTRODUCTION

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

EXECUTION CONDITION

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

19.8. USER INPUT (CVTA)

USER INPUT

.One '*'-type line per user:

!POS.!	!LEN.!	! VALUE	!MEANING	!
! 2 !	! 1 !	! '*'	!Line code	!
! 3 !	! 8 !	!uuuuuuu!	User code	!
! 11 !	! 8 !	!pppppppp!	Password	!

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

!POS.!	!LEN.!	! VALUE	!MEANING	!
! 1 !	! 1 !	! 'S'	!Transaction code	!
! 2 !	! 1 !	! 'A'	!Line code	!
! 3 !	! 6 !	! tttttt	!Table number	!
! 9 !	! 8 !	!DDMMCCYY!	Update date: beginning	!
! 17 !	! 2 !	!	!Not used	!
! 19 !	! 1 !	! '/'	!Delimiter	!
! 20 !	! 1 !	!	!Not used	!
! 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.

19.9. DESCRIPTION OF STEPS (CVTA)

CVTA: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

TABLE-CONTENTS COMPARISON: PTAV10

.Permanent input files:
-Table-description file
PAC7TD
-Error-message file
PAC7TE
-Table-contents file
PAC7TV
-User-parameter file
PAC7TG

.Input transaction file:
-Comparison requests
PAC7MV

.Output file:
-Comparison result
PAC7EX

.Output report:
-Transaction report
PAC7ET

EXTRACTION OF UPDATE TRANSACTIONS: PTAV20

.Permanent input file:
-Table-Description file
PAC7TD

.Input transaction file:
-Comparison result
PAC7EX

.Output file:
-Update transactions for use as
input of UPTA)
PAC7NU

.Output report:
-Printing of extracted transactions
PAC7ET

19.10. EXECUTION JCL (CVTA)

```
#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) CVTA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                      CVTA PROCEDURE"
echo "                      ====="
echo "Directory 'assign'      : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'       : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'     : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : DISPATCHED TABLE MANAGEMENT
# *           TABLE CONTENTS COMPARISON
# *****
. $TABDIR/assign/$1/PAC7TD.ini
. $TABDIR/assign/$1/PAC7TE.ini
. $TABDIR/assign/$1/PAC7TV.ini
. $TABDIR/assign/$1/PAC7TG.ini
PAC7MV=$PACINPUT'MBCVTA'
export PAC7MV
PAC7EX=$PACTMP'EX'
export PAC7EX
PAC7ET=$PACTMP'CVTAET.V10'
export PAC7ET
echo "Execution: PTAV10"
cobrun PTAV10
RETURN=$?
case $RETURN in
0)
    ;;
*)
    echo "Error in executing PTAV10"
    echo "Error $RETURN"
    sh $TABDIR/batch/proc/ERRPAUSE.ini
    exit $RETURN
    ;;
esac
# *****
. $TABDIR/assign/$1/PAC7TD.ini
PAC7EX=$PACTMP'EX'
export PAC7EX
PAC7NU=$PACINPUT'MVCVTA'
export PAC7NU
PAC7ET=$PACTMP'CVTAET.V20'
export PAC7ET
echo "Execution: PTAV20"
cobrun PTAV20
RETURN=$?
case $RETURN in
0)
    ;;
*)
    echo "Error in executing PTAV20"
```

```
echo "Error $RETURN"  
sh $TABDIR/batch/proc/ERRPAUSE.ini  
exit $RETURN  
;  
esac  
# *****  
echo "End of procedure"  
echo ""  
echo "Deletion of the temporary file EX"  
rm -f $PACTMP'EX'  
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	168
TABLES - EXPLOITATION & INSTALLATION		
TABLE RETRIEVAL FROM RELEASES 8.xx OR 1.2		20

20. TABLE RETRIEVAL FROM RELEASES 8.xx OR 1.2

20.1. INTRODUCTION

RETRIEVAL OF PACTABLES 8.xx OR 1.2 (RTTA) : INTRODUCTION

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

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

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

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

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

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

- . Backup in 8.xx or 1.2 format (SVTA procedure) of the TD, TV, and TG files of Release 8.xx or 1.2;
- . Retrieval in 2.5 format (RTTA procedure) of the backup produced by the preceding step, so as to produce a 2.5 backup file;
- . 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

USER INPUT

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

```
+-----+-----+-----+-----+
!Pos.! Len.! Value  ! Meaning  !
+-----+-----+-----+-----+
!  1 !   2 ! Number  ! Pivot year  !
!   !   ! other   !           !
!   !   ! than '00' !           !
+-----+-----+-----+-----+
```

20.3. RTTA: DESCRIPTION OF STEPS

RTTA: DESCRIPTION OF STEPS

BACKUP RETRIEVAL: PTAR20

.Permanent input file:

-1.2 backup file
PAC7TC

.Permanent output file:

-Temporary backup, rel. 2.5
PAC7TR

.Input file:

-user parameter line
PAC7MB

.Output report:

-Retrieval report
PAC7ET

20.4. RTTA: EXECUTION JCL

```
#!/bin/sh
#(##)
#(##)--          Release xxx Version xxx          --
#(##)
#(##)Pactables (R) RTTA BATCH Procedure
#(##)
# Parameter control
. $TABDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----"
echo "                      RTTA PROCEDURE"
echo "                      ====="
echo "Directory 'assign'          : $TABDIR/assign/$1"
. $TABDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'           : `dirname $PACTMP.`"
. $TABDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'         : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
    echo "Radical 'tmp' and 'input' files : $2"
fi
echo "-----"
echo ""
sh $TABDIR/batch/proc/MSGPAUSE.ini
# *****
# * Pactables : RETRIEVAL OF A TABLES DATABASE 8.XX OR 1.2
# *****
. $TABDIR/assign/$1/PACSAVTC.ini
PAC7TC=$PACSAVTC
export PAC7TC
PAC7TR=$PACSAVTCNEW
export PAC7TR
PAC7MB=$PACINPUT'MBRTTA'
export PAC7MB
PAC7ET=$PACTMP'RTTAET.R20'
export PAC7ET
echo "Execution: PTAR20"
cobrun PTAR20
RETURN=$?
case $RETURN in
0)
; ;
*)
echo "Error in executing PTAR20"
echo "Error $RETURN"
sh $TABDIR/batch/proc/ERRPAUSE.ini
exit $RETURN
; ;
esac
# *****
echo "End of procedure"
echo ""
echo "Calling the file TCBACKUP.CMD"
. $TABDIR/assign/$1/TCBACKUP.ini
exit $RETURN
```

VisualAge Pacbase - Operations Manual	PAGE	174
TABLES - EXPLOITATION & INSTALLATION		
COMPATIBILITY BETWEEN PACTABLES 2.5 AND VA PAC 1.6		21

21. COMPATIBILITY BETWEEN PACTABLES 2.5 AND VA PAC 1.6

COMPATIBILITY BETWEEN PACTABLES 2.5 and VA Pac 1.6

If you use Pactables 2.5 and generate table descriptions of VA Pac from version 1.6 and higher versions (or from a former version), you need using the GETA, GETD, and GETI procedures that are supplied with the Pactables 2.5 installation tape, instead of the GETA, GETD, AND GETI procedures supplied with VA Pac as they are not compatible with Pactables 2.5.

VisualAge Pacbase - Operations Manual
TABLES - EXPLOITATION & INSTALLATION
INSTALLATION

PAGE 176

22

22. INSTALLATION

22.1. INTRODUCTION

INTRODUCTION

The Pactables media pack is available in various forms; on magnetic media (tape, DAT, VDAT) or CDROM.

This media pack allows installation of the Pactables server as well as the 'dumb terminal'-type user workstations operating on UNIX (TABWST) or on WINDOWS (PACLINK). The installation of these elements is described in this chapter.

CONSTITUTION OF THE MEDIA PACK

The media pack contains the following files:

- TABINST.Vnn
- TABBASE.3PC, TABBASE.3IS, TABBASE.4PC, and TABBASE.4IS for AIX
- TABBASE.750, TABBASE.815, and TABBASE.875 for HPUX - TABBASE.ISO for other hardware
- README_E.Vnn and README_F.Vnn
- AMELIO.Vnn and CHANGES.Vnn

as well as -if relevant- the 'TABPATCH' directory, containing the files which relate to the patch(es) likely to be integrated in a re-installation.

('Vnn' represents the number of the installed release.)

CAUTION: Depending on the type of medium and UNIX system, installation files names may be entered in upper- or lowercase.

Description of TABINST.Vnn:

The TABINST.Vnn file contains the Pactables installation or re-installation commands.

These commands are written in 'shell' language.

Description of TABBASE.xxx:

The TABBASE.xxx files are compressed 'tar' files containing the Pactables execution and operations files in English and French.

On RS6000 or DPX20 systems, the operations system offers the choice of either the 'pc850' character sequences (OS/2 character set) and 'iso8859' (MS-Windows character set).

Consequently, for compatibility reasons, this choice is offered at the time of installation or re-installation.

On HP9000 machines, the software can operate with either Micro-Focus COBOL Run-Time 3.0 (HP 7.50), 3.1 (HP 8.15) or 3.2 (HP 8.75 or HP 9.xx).

On RS6000 or DPX/20 systems, the software can operate with the Micro-Focus COBOL Run-Time 3.2 or 4.0.

During installation or re-installation, the choice of a Run-Time allows you to install the files compatible with the Run-Time in use.

Description of README_F.Vnn and README_E.Vnn:

As a general rule, the README file contains information about changes in the software.

This file should be read before any installation or re-installation operation.

The README_F.Vnn file is meant for French-speaking users while the README_E.Vnn file is meant for English-speaking users.

Description of AMELIO.Vnn and CHANGES.Vnn:

These files contain a list of changes and patches offered by the new version.

The AMELIO.Vnn file is meant for French-speaking users, while the CHANGES.Vnn file is meant for English-speaking users.

Description of the 'TABPATCH' directory:

The TABPATCH contains files when relevant. Each file is named with the patch number and contains

- A list of the changes to which the patch applies
- The names and access paths of the program(s) affected by the patch
- The numbers of the versions which can use the patch.

These files are used when re-installing the software via TABINST.Vnn.

THE INSTALLATION PROCESS

It is recommended to do a general backup of the system before installing Pactables and to follow the various steps of installation very carefully:

- creating a login specific to Pactables,
- creating or modifying UNIX environment variables,
- unloading the support in the \$HOME connection directory,
- Downloading the CD-Rom in the installation directory:
"version_code"."hardware-code",
- starting up the tabinstall procedure,
- installing the TUF-TP Macro-structures,
- creating the on-line and BATCH servers,
- creating users on the network,
- installing the workstations,
- installation tests.

INSTALLATION
INTRODUCTION

PAGE

180

22
1

WARNING:

Pactables must be installed on disks physically present on the SAME UNIX MACHINE, and NOT ON NFS-mounted disks.

22.2. INSTALLING SERVERS

INSTALLING PACTABLES

CREATING A SPECIFIC UNIX USER

Most software on UNIX is installed in the specific user's account.

This is the case so as to separate the different software installed on a machine for security reasons.

DO NOT INSTALL ON THE UNIX ACCOUNT'S "root".

The UNIX administrator (root) must create a UNIX login on which Pactables will be installed.

This login will be that of the PACLAN/X-TABLE administrator.

If a Pactables database has already been installed, the same login may be used.

MODIFYING THE CONFIGURATION FILES

Connect with the Pactables administrator's login Pactables and modify the shell configuration files (.profile, .kshrc or .login).

The COBDIR, TABDIR, COBPATH and PATH environment variables will be created or modified:

```
COBDIR=<access path to COBOL Run-Time directory>
export COBDIR
TABDIR="$HOME/tablex"
export TABDIR
COBPATH="$TABDIR/tp/gnt:$TABDIR/batch/gnt"
export COBPATH
PATH="$TABDIR/bin:$TABDIR/batch/proc:$COBDIR/bin:$PATH"
export PATH
```

For the DIGITAL, SUN or TANDEM versions, create or modify the LD_LIBRARY_PATH variable:

```
LD_LIBRARY_PATH=$COBDIR/coblib
export LD_LIBRARY_PATH
```

For the RS6000 variable, create or modify the LIBPATH variable:

```
LIBPATH=${COBDIR}/coblib
export LIBPATH
```

The COBOL Run-Time here corresponds to the directory with a certain number of executables and libraries necessary for the execution of the COBOL programs.

In general COBDIR takes the value /usr/lib/cobol (refer to the document on installing the COBOL compiler and the COBOL manuals where the COBDIR environment variable is defined).

Once these modifications have been done, re-start the initialisation of the configuration by typing ". .profile" and verify that the modifications have been made by using the UNIX "set" command.

If the environment variables have not been initialized correctly, log off and reconnect to the PACTABLE administrator's account.

UNLOADING THE MAGNETIC MEDIUM

WHEN Pactables is shipped on a magnetic medium, you must unload this medium's contents on the UNIX machine, into the administrator's \$HOME directory.

This is done via the following commands:

```
cd $HOME
tar xvf /dev/...
```

('dev/...' being the path of the medium's drive)

INSTALLING THE CD-ROM

You may have to install the driver for the CD-ROM's drive in one of the system's directories.

In the CD-ROM's installation directory, go to the following sub-directory:

```
...'version_code'.'hardware_code'
```

Example with RS6000: `cd /cdrom/PT200V00.RS6`

INSTALLATION PROCEDURE START-UP:

WARNING:

To guarantee a correct installation of the software, you must provide for a disk space equal to five times the size of the TABBASExxx.Z file in the installation directory.

After downloading, in the Pactables administrator's logging directory, 2- and batch procedures

'sh tabinst* -f' for installation and operation in French,

'sh tabinst* -e' for installation and operation in English,

followed by Return or Enter.

The installation procedure includes the following steps:

- consistency check on the environment variables (positioned in .profile, .kshrc, and login files),
- display the readme information,
- display the installation menu,
- choice of the sequence of characters used (RS6000 or DPX20 only),
- choice of COBOL Run-Time used (HP9000, RS6000 or DPX20 only),
- creating the installation directory,
- assignment of the files issued from TABBASExxx,
- input of the name and creation of the test database,
- input of the TD file's access path,
- execution of the BATCH procedures to initialize the test database,
- creating the journal_install file.

Consistency check on the environment variables:

Before starting the tabinst procedure, the Pactables administrator should check carefully the changes made to the configuration file(s), even though the procedure itself carries out some consistency checks.

Displaying the information from the readme file:

To scroll through the README file use the space bar.

At the end of the display you can:

- either re-run this step,
- quit the installation procedure,
- or proceed to the next step.

Displaying the installation menu:

The installation menu is in the following form:

```
*****  
                                Pactables "version"  
*****  
  
readme      : display the readme file  
inst        : install the version  
*****  
  
Type the command name or 'x' to exit tabinst:
```

Description of commands:

- The 'readme' command reads the readme file.
- The 'inst' command starts the installation.

Type the command and follow the instructions supplied by 'tabinst'.

Choice of a character sequence (RS6000 or DPX20):

The sequence of characters is 'pc850' or 'iso8859', according to the machine's configuration (ask the UNIX administrator to find out about this configuration).

Choice of COBOL Run-Time used (HP9000, RS6000, or DPX20):

With HP9000, the COBOL Run-Time may be '750' if a Micro-Focus Run-Time 3.0 (HP 7.50 version) is used, or '815' if a Micro-Focus Run-Time 3.1 (HP 8.15 version) is used, '875' if a Micro-Focus Run-Time 3.2 (HP 8.75 or 9.xx version) is used, or '113' if a Micro-Focus 4.0 (HP 11.30 version) is used.

With RS6000 or DPX20, the COBOL Run-Time may be '3.2' if the Micro-Focus Run-Time '3.2' is used or '4.0' if a Micro-Focus Run-Time 4.x is used.

Ask your UNIX administrator to find out about the Run-Time.

Creating the installation directory:

tabinst imposes the name (tablex) on the installation directory for two reasons:

- to isolate Pactables from other applications in order to control the development more easily,
- to allow tabinst to control the type of process (installation or reinstallation) to carry out.

Assigning the files:

During this step, tabinst creates the directories described later on in this manual ('Description of created directories') and extracts the set of TABBASExxx files in the chosen language. The installation indicator -e (English) or -f (French) gives you the following elements in the language chosen at installation :

- error messages
- batch procedures

Inputing and creating the test database:

The database name must be given during installation.

It is input interactively responding to questions concerning the database (maximum 4 alphanumeric characters).

This name allows sub-directories specific to the database to be created:

- \$TABDIR/assign/"database_name"
- \$TABDIR/bases/"database_name"

- \$TABDIR/input/"database_name"
- \$TABDIR/save/"database_name"
- \$TABDIR/tmp/"database_name"

Input of the access path to the TD file:

Sites with a VisualAge Pacbase database, can use it to assign the TD file.

Executing the initialization BATCH procedures:

The test database is initialized by executing the following BATCH procedure:

- RSTA: loading the test database,

NOTE:

The COBOL Run-Time Micro Focus is needed to execute the BATCH procedures, and to start-up the BATCH and on-line servers.

Updating the journal_tabinstall file:

The journal_tabinstall file contains information resulting from the use of tabinstall. At the time of installation this information is:

- the installation date,
- the release number,
- the version number,
- the name of the sequence of characters used,
- the date of the creation of the test database.

22.3. DESCRIPTION OF CREATED DIRECTORIES

DESCRIPTION OF CREATED DIRECTORIES

The PACTABLE-UNIX installation creates a tree of directories in the under the "\$TABDIR" directory.

```
"$TABDIR"  
!           +----- "Base1"  
!           !  
!           +----- "Base2"  
!           !  
!           +----- batch_server  
!           !  
!---- assign ---+----- .....  
!           !  
!           +----- monitors  
!           !  
!           +----- tp_server  
!  
!  
!           +----- "Base1"  
!           !  
!---- bases ---+----- "Base2"  
!           !  
!           !----- ...  
!           !  
!  
!           +----- gnt  
!           !  
!---- batch ---+----- int  
!           !  
!           +----- proc  
!  
!  
!---- bin  
!  
!  
!---- cobol  
!  
!  
!---- commun  
!  
!  
!---- input  
!  
!  
!
```

INSTALLATION
DESCRIPTION OF CREATED DIRECTORIES

```
!  
!--- install  
!  
!--- method  
!  
!--- log  
!  
!--- paclink  
!  
!  
!           +---- "Base1"  
!  
!--- save   ---+---- "Base2"  
!  
!           !---- ...  
!  
!  
!           +---- "Base1"  
!  
!--- tmp     ---+---- "Base2"  
!  
!           !---- ...  
!  
!  
!           +---- gnt  
!  
!--- tp      ---+---- int  
!  
!           +---- save  
!  
!  
!           +---- "User1"  
!  
!--- users   ---+---- "User2"  
!  
!           !---- ...
```

THE SYSTEM DIRECTORIES

The \$TABDIR/bin directory

The "bin" directory contains the set of C executable files: BATCH monitors (tabbatch), TP (tabtp), the UNIX workstation (tabwstsock), as well as the conversion utilities.

The \$TABDIR/tp directory

The "tp" directory contains the COBOL executables ("int" and "gnt" directories), as well as the transaction backup files ("save" directory), allowing the operation of the on-line server(s).

```
tp +--- gnt - on-line programs (compiled code)
!
!--- int - on-line programs (interpreted code)
!
!--- save - "serveur1".ts (transaction
- "serveur2".ts backup)
:
```

The \$TABDIR/batch directory

The "batch" directory contains the COBOL executables ("int" and "gnt" directories), and the shell commands for the operation of a PACTABLE database ("proc" directory).

```
batch +--- gnt - BATCH programs (compiled code)
!
+--- int - BATCH programs (interpreted code)
!
+--- proc - BATCH procedures
```

All the BATCH procedures are described in this manual.

The \$TABDIR/install directory

The install directory contains: the tabinstall procedure which allows a new database to be created and servers to be created or destroyed, files which allow a new database or server to be initialized, as well as the journal_tabinstall file which memorizes all the installation operations.

The \$TABDIR/cobol directory

This directory contains the cobol source for the table access sub-programs as well as the cobol source for the TD and TV files loading programs.

The \$TABDIR/method directory

This directory contains the VA Pac update transaction file used for macro-structures in the TUF-TP facility.

THE SPECIFICATION DATABASES DIRECTORIES

These directories contain as many subdirectories as there are specification databases; each subdirectory having the name of the corresponding database.

The \$TABDIR/assign/"database_name" directory

The "assign" directory has a subdirectory for each database installed. Also found here is the set of command files containing the assignment of standard installed files in the TABDIR/bases and \$TABDIR/bases/"database_name" directories. These command files are called PAC7TE.ini, PAC7TG.ini, ... for the database files.

The assign directory also has three subdirectories:

- batch_server,
- tp_server,
- monitors.

The batch_server and tp_server directories contain the server's or servers' initialization files. The monitors' initialization files are in the monitors directory.

The \$TABDIR/bases directory

The databases directory contains the error messages file which can be used by the set of installed databases.

The \$TABDIR/bases/"database_name" directory

The database directory contains a subdirectory for each database installed, in which the database indexed files (TV and TG) appear. The table descriptions file (TD) is in the PACLAN/X specification database.

The \$TABDIR/save/"database_name" directory

The save directory contains a subdirectory for each database installed, which contains the sequential backup for the TC database.

Also in this directory are the command files ensuring a rotation on two versions of the database sequential backups (TCBACKUP.ini file).

The \$TABDIR/input/"database_name" directory

The input directory contains a subdirectory for each database installed which contains all the input transaction files of the MBxxx BATCH procedures (MBPRTA, MBLDTA, MBRETA ...) as well as the transaction files created by the extraction utilities coded. MVxxxx (MVEXTA, MVCVTA, ...).

The \$TABDIR/tmp/"database_name" directory

The tmp directory contains a subdirectory for each database installed. It contains the temporary files as well as the BATCH procedures execution reports.

The \$TABDIR/log directory

The log directory contains a trace file for each server if a trace is active at the on-line monitor level.

THE SHARED DIRECTORIES

They are only for the Pactables installation set and are created by the tabinstall procedure.

The \$TABDIR/commun directory

The batch communication files TLB and TBD and the on-line server communication files TCM and TCD are in the "commun" directory.

The TLB file memorizes the content of the batch requests. The TBD er TBD mirrors the state of these requests.

These files are prefixed by the name of the database concerned.

The TCM file contains the messages between the on-line server and the workstations.

The TCD file contains information linked to the screens, exchanged between the workstations and the on-line server.

These files are prefixed by the name of the server concerned.

The \$TABDIR/users directory

The "users" directory includes the results of the work started up by each user (procedure to print the requests submitted on-line). This distribution is done in n sub-directories one per user.

```
example:      users      +--- JAMES --- nnnnn.inp
                !                nnnnn.sta
                !                etc
                !--- PETER --- ppppp.inp
                !                ppppp.sta
                !                etc
```

where nnnnn (resp. ppppp) is the job number (see 'batch monitor screen), "inp" and "sta" are examples of files created by the BATCH server.

WINDOWS 'DUMB' WORKSTATIONS DIRECTORY (PACLINK)

The following files are created in the TABDIR/paclink directory:

- . PACLINK.EXE: user workstation monitor
- . GSTCPIP.EXE: communications management by the "Windows Socket" interface
- . PACLINK.PRM: parameters file of the workstation monitor
- . GSTCPIP.PRM: communications management parameters file in the case of communication by "Windows Socket" interface.
- . GSHUNIX.TAB (for HPUX), GSAS850.TAB (for AIX/PC850), GSANSI.TAB (for AIX/ISO8859, SunOS, IRIX, or OSF1): characters transcodification table,
- . VAPAC.FON: font used when activating a screen trace in GSPACLAN.EXE.

22.4. *INSTALLING TUF-TP MACRO-STRUCTURES*

INSTALLING TUF-TP MACRO-STRUCTURES

To run TUF-TP, Macro-structures need to be integrated in VA Pac.

These Macro-Structures are provided at installation on the method files directory (\$TABDIR/method), in the MBUPDT.TUF file.

These batch transactions are to be entered in the VA Pac database through the PROCUPDT procedure which is used to update the network.

The PROCUPDT batch procedure takes transactions in input in a MBUPDT file under the \$PACDIR/input/"database_name" directory.

The file located on the method directory must be copied in MBUPDT on the \$PACDIR/input/"nom_base" directory.

Once you have chosen the library in which the entities will be loaded, modify the MBUPDT file with an editor by filling the '*' line at the beginning of the transactions and start the UPDT procedure then.

22.5. CREATING/DELETING SERVERS

CREATING/DELETING SERVERS

The "tabadmin" procedure in the \$TABDIR/install directory offers a menu which contains the commands for creating or destroying the on-line and BATCH servers.

Creating an on-line server:

The UNIX system files must be updated first when creating BEFORE the "tabadmin" procedure which creates an on-line servers is executed.

Updating the system files:

The communication port number must be reserved in the UNIX "/etc/services" file in order for the "socket" communication interface to be recognized (whether it is used or not).

A communication port number must be chosen for each on-line server. This choice must be made knowing that a certain number of ports are reserved for standard "internet" network applications or for other applications. Other ports are allocated dynamically.

It is therefore necessary to chose a port number that is not reserved in "/etc/services" and which is not currently being used, i.e. not being allocated dynamically.

Example of use:

A port number must be chosen for each on-line server. If port number 1500 is chosen

- . the port number 1500/tcp must not already be in the "/etc/services" file;
- . the port number must not be currently in use. The following UNIX command ensures this:

```
netstat -a \ grep 1500
```

- . if the first two conditions are verified the UNIX administrator (root) must update the "/etc/services" file by the following line:

```
tablex 1500/tcp
```

The port number chosen for each on-line server must be carefully noted as this information is needed by the "tabadmin" procedure when creating the on-line server.

Starting up the "tabadmin" procedure:

To activate the tabiadmin procedure it is necessary to be in the TABDIR/install directory with the help of the command:

```
"cd $TABDIR/install"
```

and to start the command:

```
"sh tabadmin -f" to install in French,
```

```
"sh tabadmin -e" to install in English,
```

followed by pressing the return key.

Description of steps:

The steps of the procedure to create a server are as follows:

- . displaying the menu (below),
- . inputting the command
- . inputting the database name (if creating),
- . inputting the server name,
- . inputting the number of workstations (if creating an on-line server),
- . inputting the communication port number (if creating an on-line server),
- . updating the journal_tabinstall file.

Displaying the menu:

The menu for creating or destroying a server is as follows:

```
*****  
                                     Pactables "version"  
*****  
  
.  
.  
cresertp      : creating an on-line server  
creserbatch   : creating a BATCH server  
delsertp      : destroying an on-line server  
delserbatch   : destroying a BATCH server  
.  
.  
*****  
  
Type the command name or 'x' to exit tabadmin:
```

Input the command and follow the instructions supplied by 'tabadmin'.

Updating the journal_tabininstall file:

Add the information concerning creating and destroying a server to the journal_tabininstall file:

- server's name and date of creation or destruction.

22.6. CREATING NETWORK RESOURCES

CREATING NETWORK RESOURCES

COMMUNICATION THROUGH SHARED FILES

Files used for communication between the workstations and the Pactables-Unix servers as well as PRTA output files, should always be accessible for update by workstations and Pactables-Unix servers.

These files must be physically installed on the UNIX system where Pactables is installed, and must be declared as shareable resources on the local network.

It is recommended to define two virtual disk drives on workstations:

- . One in the \$TABDIR/commun directory (used to access and update the communication files by the on-line and BATCH servers and the workstations),
- . One in the \$TABDIR/users directory (used by the BATCH servers to write the results of generation-print requests and by the workstations to read these files).

The installation of these virtual disk drives depends on the network software in use. For more information refer to the product's documentation on the distribution of the resources (UNIX tree) on the server and how to use these resources on the WINDOWS and OS/2 workstations.

Example (LAN MANAGER-type network software)

Under username 'pacuser', the user must access the \$TABDIR/ users directory on the Pactables server whose name is 'SERVER'.

You must first create a resource for this directory on the UNIX machine (login = root or login of the network administrator):

```
net share PACUSER=$TABDIR/users/pacuser
```

And to be able to access this resource from a DOS or OS/2 workstation, create a new logical disk unit, "I:" for example:

```
NET USE I:\SERVER\PACUSER
```

Then, the user can directly access the generation-print output from a WINDOWS or OS/2 workstation under the 'I:\pacuser' directory.

You must also share the database communication file directory on the network. Enter the following command on the server:

```
net share COMMUN=$TABDIR/commun
```

and the following command on the workstation(s):

```
net use J:\SERVER\COMMUN
```


COMMUNICATION VIA THE SOCKET INTERFACE

The socket communication interface is based on the TCP/IP network protocol. Besides the installation itself, the implementation of TCP/IP requires some essential files to be configured.

Each machine's "hosts" file must contain the IP addresses of the machines with which communication is established.

The 'ping' command allows you to check whether the TCP/IP network is in working order.

Example: ping "host_name"

If it is, the PC workstations and the Pactables on-line server are enabled to communicate via the socket interface.

For users to be able to access the generation-print files, some specific NSF resources must be defined or the FTP must be used.

The principle is to declare the \$TABDIR/users directory as a resource which can be 'exported' by the NSF server (UNIX machine), and to 'mount' this resource as a read-write network resource on the client NSF workstation (Windows PC).

(NFS is a standard TCP/IP product for UNIX systems as well as PC systems. For further details on the creation of specific NSF resources, refer to the NSF documentation.)

22.7. *INSTALLING WINDOWS DUMB-TERMINALS*

INSTALLATION OF A WINDOWS 'DUMB' TERMINAL (PACLINK)

Once the Pactables servers are installed, you may start installing the 'dumb terminal' WINDOWS or WIN-OS/2 workstations.

1. COPYING THE PACLINK FILES

All the PACLINK specific files are found in the \$TABDIR/paclink directory. (These files are described in Chapter 'Installation', Subchapter 'Description of created directories'.

They must be copied on the workstation via the local network in an installation directory chosen by the user.

2. UPDATING THE PARAMETER FILES

Two parameter files should be modified according to the installation characteristics:

GSTCPIP.PRM

Specify the following information:

- . The name of the UNIX machine (UNIX hostname command) on which Pactables is installed (P0 line), or the IP address of the machine (P1 line).

Warning: P0 and P1 are exclusive lines.

- . The socket number given for Pactables TP server (P2 line).
- . The file used when exiting Pactables while saving the conversation (SA line).
- . The file used for the Trace function when the function is activated (D0 line).

Example: on a UNIX system with "MYHOST" as network name, and with the on-line server 'listening' on socket 1500, the dialogue backup file: C: TABBASE SAVE.TXT and the trace file C: TABBASE GSPACLAN.DEB, you have:

P0 MYHOST

P2 1500

SA C: TABBASE SAVE.TXT

DO 0 1 0 1 0 0 0 C: TABBASE GSPACLAN.DEB

PACLINK.PRM

Specify the disk drive and directory where GSTCPIP.EXE (G line) and GSTCPIP.PRM (E line) are installed.

Example: G C:\TABBASE GSTCPIP.EXE
 E C:\TABBASE GSTCPIP.PRM

CREATING A WINDOWS ICON

Create a program: 'File' menu, 'Shortcut' command, in a group of your choice. The 'command line' must read the complete access path to the PACLINK.EXE file, separated by a blank from the PACLINK.PRM file access path.

Example: C: TABBASE PACLINK.EXE C: TABBASE PACLINK.PRM

22.8. INSTALLING UNIX DUMB-TERMINALS

INSTALLING A UNIX 'DUMB' WORKSTATION (TABWSTSOCK)

The 'tabwstsock' executable which is installed in the TABDIR/bin directory, ensures the start-up of the 'dumb' UNIX workstation.

'tabwstsock' communicates with the on-line server via the 'socket' interface.

REQUIREMENTS FOR USING A UNIX 'DUMB' WORKSTATION

In order to use a UNIX 'dumb' workstation, you must:

- Create a user account and a specific UNIX login on the on-line/batch server.
- Check the activation status of the user ID bit in the 'tabwstsock' executable file, using the following commands:

```
cd $TABDIR/bin  
ls -l tabwstsock
```

The system displays two lines, whose first four characters are: '-rws'.
If it does not, the Pactables administrator must connect under his/her own login account and issue the following command:

```
chmod u+s tabwstsock
```

- Login to the Pactables installation user account and modify the shell configuration files (.profile, .kshrc or .login according to the case) in order to create or modify the TABDIR and PATH environment variables:

```
TABDIR=<access path to the Pactables directory>  
export TABDIR
```

```
PATH=$TABDIR/bin:$PATH  
export PATH
```

After this operation, restart the configuration initialization by typing '.profile', and, using the UNIX command 'set', check that the modifications were taken into account in the configuration files.

If the environment variables are not correctly initialized, log out and then log in again to the user account.

STARTING THE UNIX 'DUMB' WORKSTATION

Please refer to chapter "MONITOR START-UP", subchapter 'PACLINK: UNIX DUMB WORKSTATION START-UP'.

22.9. UTILIZATION TESTS

UTILIZATION TESTS

These test jobs include the following steps:

- . Tests on on-line use,
- . Test on BATCH updating, printing and reorganization,
- . Test on table generation.

The test deck includes three tables:

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

Tests on the user parameters transaction:

Submit the TABPARM parameter. The user code defined in the test database is the manager's: '*****' user code and 'SUPER' password. Use this code for the transaction, to define other users, and to modify the database manager's user code password.

Pactables on-line tests:

Submit an on-line server; connect it to a workstation.
Consult all the screens.
Execute the updates.

BATCH tests:

Execute the PRTA procedure.
Execute the EXT A procedure.

Close the on-line server.

Execute the UPTA procedure.

Test table reorganization:

- . Backup TV and TD (copy under different names).
- . Execute the reorganization procedure (RETA) which

includes:

- Initializing (PTA20A Prog.) TW and TS files,
- Reorganizing TV and TD into TW and TS,
- Renaming (or recopying) TW and TS to TV and TD,
- . Verify the files through a printing

procedure (PRTA).

- . Submit the on-line server and a station.

After reorganization carry out some on-line tests.

Table generation test (GETT procedure):

- . Close the station and the on-line server.
- . Execute an extraction procedure (GETA or GETD).
- . Execute the GETT procedure.
- . Verify the execution.
- . Start-up the on-line server and a workstation. Do some verification tests.

22.10. MANAGING SEVERAL PACTABLES BASES

MANAGING SEVERAL PACTABLES DATABASES

Standard location of a database's files:

During a standard installation, one of the parameters of the tabadmin procedure corresponds to the database name (test for example). Sub-directories specific to the database may be created:

```
$TABDIR/assign/test  
$TABDIR/bases/test  
$TABDIR/tmp/test  
$TABDIR/save/test  
$TABDIR/input/test
```

Test database files are installed in each of the above directories.

The TE messages file is installed directly in TABDIR/bases as this file can be common to several Pactables databases.

The server and BATCH procedures assign files located in the databases directories and in bases/"database_name" via the commands file intermediary created at installation in assign/"database_name" which make standard installation modifications easier.

The standard Pactables architecture is capable of managing a multi-database environment.

Creating a new database

To create a new database, go to the \$TABDIR/install directory by using the following command:

```
cd $TABDIR/install
```

Type the command:

```
-"sh tabadmin -f" to create in French,
```

```
-"sh tabadmin -e" to create in English,
```

and press the Enter key.

Description of steps:

The steps of the procedure to create a database are as follows:

- displaying the menu (below)
- inputing the command (crebase),
- inputing the database name,
- creating directories specific to the database,
- writing information concerning the new database in the journal_tabinstall file.

Displaying the menu:

The tabadmin menu is as follows:

```
*****  
                                Pactables "version"  
*****  
  
      .  
      .  
crebase      :   creating a new database  
      .  
      .  
*****  
  
Type the name of the command or 'x' to exit tabadmin.
```

Input the command and follow the instructions supplied by 'tabadmin'.

Updating the journal tabinstall file:

Add the information concerning the creation of a database to the journal_tabinstall file:

- database name and date of creation.

Initializing the database

"tabadmin" initializes the new database with the test database TC file.

22.11. REINSTALLING PACTABLES

RE-INSTALLING PACTABLES

Pactables must be re-installed on receiving a version with corrections or new developments.

The new sub-version, identified by a number, contains:

- . the software,
- . a list of corrected anomalies,
- . re-installation instructions.

In general, only the system files (TE, the \$TABDIR/method transaction files, the generation skeletons) and the programs (BATCH and on-line) are affected by the new sub-version.

GENERAL COMMENTS:

- . The re-installation procedure does not create directories which are expected to be identical to those created during the first installation of the version.
- . It does not copy the BATCH procedures (\$TABDIR/batch/proc directory) if the database management has modified them to adapt them to the site, except if the the new sub-version will not run with the former procedures.

STARTING THE RE-INSTALLATION PROCEDURE:

After unloading the support files, as for an installation (refer to Subchapter 'Installing Pactables'), in the in the connection directory, type the command (identical to that for installation):

"sh tabinst* -f" to reinstall in French,

"sh tabinst* -e" to reinstall in English,

and press the Return key.

Description of steps:

The steps of the re-installation procedure are as follows:

- consistency check of the environment variables (.profile, .kshrc and .login files),
- displaying the information from the readme file,
- displaying the re-installation menu,
- choice of the sequence of characters (only on RS6000 or DPX20),
- choice of COBOL Run-Time used (only on HP9000, RS6000 or DPX20),
- assignment of files issued from TABBASExxx,
- updating the journal_tabinstall file.

Consistency check of the environment variables:

See Subchapter "Installing Pactables".

Displaying the re-installation menu:

The re-installation menu is as follows:

```
*****  
                                Pactables "version"  
*****  
  
readme      : displaying the readme file  
patch       : selective re-installation of patches  
reinst      : re-installation of the version  
*****
```

Type the command name or 'x' to quit tabinst:

Description of commands:

- The 'readme' command: to consult the readme file.
- The 'patch' command: to re-install selectively the improvements of the new version.

The tabinst procedure displays the list of patches of the version on the screen and for each patch the list of improvements concerned. The user inputs the patch(es), separated by a space.

For more details on the improvements, the user can consult the lists supplied with the version.

- The 'reinst' command: to completely reinstall the new version.

Input the command and follow the instructions supplied by 'tabinst'.

Choice of the sequence of characters (RS6000 - DPX/20):

See Subchapter "Installing Pactables".

Choice of COBOL Run-Time used (RS6000-HP9000):

See subchapter "Installing Pactables".

Updating the journal_tabininstall file:

Add the information concerning re-installation to the journal_tabininstall file.

- the re-installation date,
- the release number,
- the version number,
- the name of the character sequence used.

The following steps are carried out by the Pactables administrator.

Re-run the BATCH procedures:

In general, the BATCH procedures are re-run manually by the administrator, they are updated via an editor or copied.

All of the procedures of the new version are unloaded in the
\$TABDIR/batch/proc.'version_number' directory.

If the new version will not run with the former procedures, they are saved in the
TABDIR/batch/proc.old directory and the new ones are loaded in the
\$TABDIR/batch/proc directory.

The administrator must modify the new BATCH procedures in the \$TABDIR/batch/proc directory to adapt them to the site.