VisualAge Pacbase

Pactables - UNIX

Version 3.5

VisualAge Pacbase

Pactables - UNIX

Version 3.5

Contents

Notices

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

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

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

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

v

Trademarks

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

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

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

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

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

Chapter 1. Foreword

How to use this manual for system installation

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

Chapter 2. Pactables components

Introduction

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

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

Note:

Pactables is installed independently of the other VisualAge Pacbase functions.

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

The macro-structures of TUF-TP facility

- The Macro-structures are the following ones:

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

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

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

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

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

The 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/system/gnt directory)
- A file containing the Pactables error messages and Help documentation: TE

Characteristics	Value
Size	approx. 900 records approx. 1,2 mega-Bytes
Organization	Indexed
Recsize	90
Key	17 (position 1)
Utilization	Batch and on-line
Location	$TABDIR/system/skel
Internal Name	PAC7TE

Comment

$TABDIR is an installation parameter.

The 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)

Characteristics	Value
Organization	Indexed
Recsize	240
Key	21 (position 1)
Utilization	Batch and on-line
Location	$TABDIR/data/"database_name"/base
Internal name	PAC7TD

The table contents file (TV)

Characteristics	Value
Organization	Indexed
Recsize	80 to 1100
Key	35 (position 5)
Utilization	Batch and on-line
Location	$TABDIR/data/"database_name"/base
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)

Characteristics	Value
Organization	Indexed
Recsize	85
Key	22 (position 1)
Utilization	Batch and on-line
Location	$TABDIR/data/"database_name"/base
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)

Characteristics	Value
Organization	Sequential
Recsize	1061
Utilization	Batch
Location	$TABDIR/data/"database_name"/save
Internal name	PAC7TC

Comments on the location of the TD file

For sites using VisualAge Pacbase and Pactables, the TD file is located in the VA Pacbase specifications database. The VA Pacbase and Pactables installation procedures assign the TD file in this directory via the PAC7TD.ini files (located in the VA Pacbase and Pactables 'assign' directory).

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

For sites using 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

User access facilities

Some sub-programs enable user applications to access Pactables:

Code	Meaning
BVPTA800	Optimized access function
BVPTA900	Generalized access function

Code	Meaning
BVPFT00	TUF facility
BVPFT90	TUF facility

These programs are available in the '.gnt' form, compiled in the Pactables UNIX environment which can be executed with the same Micro Focus or AcuCobol Run-Time.

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

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

Chapter 3. Installation

Prerequisites

Hardware and software

- Architecture: An UNIX server
- Memory: RAM memory (64 Mb). You may need additional memory according to the number of servers installed on the same machine.
- Software - Cobol Runtime:
  - For the MICROFOCUS version:

    MICRO FOCUS Application Server
  - For the Acucorp version:

    ACUCOBOL-GT
  - CAUTION:

    You must first install the Cobol runtime and update the system environment variables (PATH, COBPATH...).

    For AcuCobol, the PATH variable must be completed with the path of the AcuCobol runtime (usually ..\AcuGT\bin). The machine must then be rebooted.
- Installation support: CD-ROM driver.

Disk space

The disk space used by the files depends on the size of the applications managed by the system.

The disk space required for the servers installation is around 6.5 millions of bytes.

System installation

General overview

Pactables is provided on a CDROM which contains the following elements:

- TABINST.Vnnx,

  where "Vnnx" is the number of the installed version.
- TABBASE.xxx,

  where "xxx" is the version of the used compiler.

  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.Vnnx 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.

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

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:

- update of the "/etc/services" file,
- creation of a specific login to the software,
- creation or modification of the environment variables,
- installation of the system, from the CD-ROM,
- installation of the Pactables database.

Between the system installation and the Pactables database installation, the Mklink must be executed. If an error occurs, you must correct it, "link" the executable programs to the runtime, if necessary. Then restart the installation of the databases by executing the "tabadmin" procedure (refer to the "Repository installation" section in this chapter) located in the TABDIR/system/install directory.

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

For the Microfocus or Acucorp versions, the COBOL runtime is indispensable for the product operation. It must be installed before the Pactables installation.

CAUTION:: For the installation of the Acucobol version on HP-UX, you must modify the [nflocks] system variable ("Max Number of File LOCKS") and give to it a value higher than 9000.

Update of the "/etc/services" file

The use of the 'socket' communication interface for the communication between the server and the clients requires the reservation of communication port numbers in the '/etc/services' UNIX file.

This file must be updated BEFORE the installation of the Pactables databases. Since this installation can follow the system installation, this file must be updated BEFORE the system installation.

A communication port number must be associated with each database. Communication ports cannot be chosen at random.

Some ports are reserved for standard 'internet' applications or for later applications. Other ports are allocated in a dynamic way.

So you must choose a port number which is not used in '/etc/services' and which is not being used, i.e. dynamically allocated.

Example:

A port number must be chosen for each on-line server. Let us suppose that we have chosen number 52040.

- The port number 52040/tcp must not be in the '/etc/services' file ;

- The port number must not be currently used. You can make sure of that via the following UNIX command:

```
netstat -an | grep 52040
```

- If both these conditions are met, the UNIX (root) administrator must update the "/etc/services" file using the following line:

```
tablex 52040/tcp
```

The port number chosen for each on-line server must be carefully noted because it will be needed when creating Pactables databases.

Creation of a specific UNIX user

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

This is necessary for security reasons, in order to isolate the various software programs installed on one machine.

NEVER INSTALL UNDER THEìROOTüUNIX ACCOUNT!

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

This login will be that of the Pactables administrator.

Modification of the configuration file

Log in with the Pactables administrator login, and update the shell configuration files (.profile, .kshrc or .login).

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

```
TABDIR="$HOME/tablex"
export TABDIR
PATH="$TABDIR/system/bin:$TABDIR/system/proc
     :$COBDIR/bin:$PATH"
export PATH
```

The environment variables COBDIR, COBPATH, ACUDIR and CODE_PREFIX must be added, according to the runtime:

For Microfocus:

```
COBDIR=<path to COBOL Run-Time directory>
export COBDIR
COBPATH="$TABDIR/system/gnt"
export COBPATH
```

For AcuCobol:

```
ACUDIR=<path to COBOL Run-Time directory>
export ACUDIR
CODE_PREFIX="$TABDIR/system/acu"
export CODE_PREFIX
```

If the TMPDIR variable (temporary directory of the Cobol Run-Time) is assigned, the indicated directory must exist.

For COMPAQ/TRUE64 (DIGITAL), SUN, HP-UX or LINUX version of Microfocus, create or modify the LD_LIBRARY_PATH variable:

```
LD_LIBRARY_PATH=/usr/lib:$COBDIR/coblib:$TABDIR/system/bin
   export LD_LIBRARY_PATH
```

If you use Server Express, LD_LIBRARY_PATH must contain:

```
LD_LIBRARY_PATH=/usr/lib:$COBDIR/lib:$TABDIR/system/bin
   export LD_LIBRARY_PATH
```

If you use Acucobol, LD_LIBRARY_PATH must contain:

```
LD_LIBRARY_PATH=/usr/lib:$ACUDIR/lib:$TABDIR/system/bin
   export LD_LIBRARY_PATH
```

For the AIX version, you must use the LIBPATH variable instead of LD_LIBRARY_PATH.

For the HP-UX version, in addition to the LD_LIBRARY_PATH variable, you must set the SHLIB_PATH variable as following:

```
export SHLIB_PATH=$TABDIR/system/bin:$SHLIB_PATH
```

The Cobol Run-time corresponds to the directory which contains the executable files and libraries required to execute Cobol programs.

For Microfocus, COBDIR can take the following values: /usr/lib/cobol, /usr/lpp/cobol or /opt/cobol (refer to the installation documentation of your COBOL compiler and to your COBOL manuals for further details on the COBDIR environment variable).

For Acucobol, The ACUDIR variable contains the installation directory of the compiler or of the COBOL Run-time.

Once you have performed these updates, 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, in the Pactables administrator's account.

Code page definition

The Repository code page is an IBM-923 code page equivalent to the ISO8859-1 code page which supports the Euro character (ISO8859-15).

Positioning the 'LANG' code for the UNIX user

In order to visualize properly the reports which contain national characters, the user must define a code page compatible with the Pactables Repository code page.

Refer to your documentation related to the UNIX system to define a code page compatible with the ISO8859-1 or ISO8859-15 code page for the Euro character support.

As a general rule, the positioning of the 'LANG' variable alone is sufficient to define the code page to be used.

Example : for the support of French national characters, you set:
- on AIX, SOLARIS, OSF1 :
  ```
  export LANG=fr_FR.ISO8859-1
  ```

- on HP-UX :
```
export LANG=fr_FR.iso8859-1
```

Installing from a CDROM

If necessary, mount the CDROM driver on a system directory.

Installation procedure start-up

To ensure a correct installation of the software, check that the disk space is equal to five times the size of the TABBASExxx.Z file in the installation directory.

After downloading, in the Pactables administrator's logging directory, or the CDROM mounting, go to the following directory:
```
PT350"hardware_code"."version_code"
```

which is located in $HOME or in the CDROM mounting directory.

Example on AIX:
```
cd $HOME/PT350AIX.Vnn (nn = version number)
```

and type the following command:
```
  sh tabinst.Vnnx            (nn = version number,
                              x = subversion number)
```

followed by a carriage-return (Enter or Return key).

The installation is made up of the following steps:
- Consistency check on the environment variables,
- Display of the installation menu,
- Selection of the COBOL run-time in use,
- Selection of the generation language code,
- Creation of the installation directory,
- Distribution of files coming from TABBASE.xxx,
- Inputting and creating the test database,
- Creation of the journal_tabinstall file.

Consistency check on the environment variables

Before running the TABINST.Vnnx procedure, the Pactables administrator must check the modifications done in the configuration file(s), even though the TABINST.Vnnx procedure performs some consistency checks.

Display of the installation menu

The installation menu looks like this:

```
************************************************************
                  Pactables
************************************************************
 inst         :   installation of the version
************************************************************
Please enter the command name or 'x' to exit from
TABINST.Vnnx
```

Description of commands:
- The 'inst' command starts the installation process.

Enter the command and follow the instructions given by TABINST.Vnnx.

Creation of the installation directory

This proposition is made on some platforms at installation.

Ask your UNIX Administrator to know which run-time is used.

Creating the installation directory:

TABINST.Vnnx requires the installation directory to be named tablex in order to:
- isolate Pactables from the other applications and so control its evolution more easily,
- allow TABINST.Vnnx to control the type of processing (installation or reinstallation) to carry out.

Distribution of files:

During this step, TABINST.Vnnx creates the directories described further on in this manual ('Description of created Directories') and extracts all the TABBASE.xxx files.

Input and creation of 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/config/"database_name"

- $TABDIR/data/"database_name"/base
- $TABDIR/data/"database_name"/save
- $TABDIR/data/"database_name"/script
- $TABDIR/data/"database_name"/tmp
- $TABDIR/data/"database_name"/users

<u>Update of the journal_tabinstall file</u>

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 date of the creation of the test database.

List of installed elements

The installation copies:
- the translation file, the labels file produced by the procedures in case of error or for information,
- the programs,
- the procedures,
- the database creation utility,
- the startup script models.

When the installation is finished, the following directories are created:
- $TABDIR/config,
- $TABDIR/data,
- $TABDIR/system.

An element of the System: the online server

Configuration

When started, each listener executes a 'BVPSERVER.ini' command file type to assign any environment file it needs.

This file is created when the listener is created in the TABDIR/config/ "database_name" directory and contains the following environment variables:

```
- BVPSOCKET      : socket number,
- SRV_DIR        : directory containing the listener
                   trace files.
```

Startup

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

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

To carry out the operations above the commands interpreter of the listener (tabtp) is 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 listener commands can be inserted into the commands file.

For example, the com_table file contains the following commands:

```
# display the listener reports
  tabtp info
# start-up the PTST listener
  tabtp start PTST
# display the information on PTST listener for ten seconds
  tabtp info PTST
  sleep 10
# start up the 003 station purging command
# depending on the PTST listener
 tabtp purge PTST 003
```

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

The available commands are as follows:

```
    -debug      (activation/deactivation of the debugger),
    -exit       (exit "shell" mode),
```

```
-help        (help on the syntax of a command),
-info        (information on the listener(s)),
-purge       (purge a station),
-purge_server (purge the listener),
-shutdown    (stop the listener without confirmation),
-start       (start the listener),
-stop        (stop the listener with confirmation),
-view        (display the listener'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.

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

To do so, refer to the Chapter 'Installation', Subchapter 'System installation', Section 'Modification of the configuration file'.

Description of the commands
Debug command

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

There are two types of trace result files:
```
srv[process_number].txt
```

to trace the listener processing in touch with new connections.
```
dial[process_number].txt
```

to trace the processing of each connection to the listener.

And so to each connection to the listener corresponds a trace.

These files are located in the SRV_DIR directory. SRV_DIR is an environment variable assigned in the "$TABDIR/config/[database_name]/BVPSERVER.ini" configuration file. Its default value is TABDIR/data/[database_name]/tmp/ server".

EXAMPLE:

- Start the debug mode on the PTST listener

```
tabtp debug PTST on        (in command mode)
Pactables : debug PTST on  (in shell mode)
```
- Stop the debug mode on the PTST listener

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

- Different trace levels can be applied:

1- Level 1

Minimum trace to follow the listener processing with the calls to the COBOL communication monitor.

2- Level 2

Detailed trace of the listener processing.

3- Level 4

Trace of messages exchanged between the listener and the client.

The 'debug on' command activates a level 1 trace on the active listener. To use other traces, the listener must be restarted after setting SRV_TRACE variable in the "BVPSERVER.ini" file.

EXAMPLE :

SRV_TRACE=1 for a level 1 trace

SRV_TRACE=3 for a level 1 and 2 trace

SRV_TRACE=5 for a level 1 and 4 trace

The 'debug off' command stops the trace creation for the new listener connections.

Exit command

This command is used to exit the shell mode (commands interpreter) previously done by tabtp -s.

Help command

This command displays help concerning a listener administration command. If no parameter is passed, the list of available commands is displayed. If a particular command is specified, the help of this command is displayed on the screen.

EXAMPLE: request help on the syntax of the start command

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

Info command

This command displays the information on the listener(s).

This command followed by a database name (info 'database_name') gives the following information:

```
- Active        (if the listener is active),
- Inactive      (if the listener is inactive),
- Error         (if the listener is stopped abnormally).
```

Purge command

This command purges a workstation (in other words it disconnects a workstation).

If the interpreter (tabtp) is in command mode, the syntax of the purge command is as follows:

```
'tabtp purge <base_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 listener's name, after the use of the view command, for example).
- 'purge <base_name> <no. of station to be purged>' if the prompt is "Pactables:".

Purge_server command

This command purges a listener in the case of abend, displayed with the 'Error' status after the 'info' command. The syntax for the purge_server command is as follows:

```
'tabtp purge_server <database_name>'.
```

In shell mode the syntax for the command is:

```
'purge_server <database_name>'.
```

N.B.: This command deletes the listener process(es), as the used IPC resources (semaphores, shared memory).

Shutdown command

This command stops the listener. The database's name must be passed as a parameter of the command.

EXAMPLE: stop the PTST listener

```
tabtp shutdown PTST        (in command mode)
Pactables: shutdown PTST   (in shell mode)
```

Start command

This command starts up the listener. The database's name must be passed as a parameter of the start command. The initialization file BVPSERVER.ini must be in the TABDIR/config/'database_name' directory. The listener start-up program, tablaunch, is launched and starts the 'tabserver' process. If problems arise (listener lock), in a first time use the command:

```
tabtp purge_server 'listener_name'
```

If this command has no effect, the 'tabserver' process can always be killed using the 'kill -15' command or 'kill -9' followed by the process number (PID).

EXAMPLE: start-up the PTST listener

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

Stop command

This command stops a listener. The database's name must be passed as a parameter of the stop command. You have to confirm stopping the listener.

EXAMPLE: stop the PTST listener

```
tabtp stop PTST        (in command mode)
Pactables: stop PTST   (in shell mode)
```

View command

This command displays information concerning a listener on a database: the list of connected stations, the IP address of the client workstation, the PID number of the client process launched by the listener, the name of the executed COBOL program and its 'Elapse' execution time in milliseconds. It also gives information on the status of the semaphore used to synchronize the concurrent accesses to the database.

EXAMPLE: display information about the PTST listener

tabtp view PTST (in command mode)

Pactables: view PTST (in shell mode)

Repository installation

Pactables database

More than one Pactables database can be installed ; each has its own environment.

A Pactables database can be installed right after the system installation, or later with the 'tabadmin' procedure which is located in the directory:

$TABDIR/system/install.

This installation is constituted of the following steps:

Display of the installation menu

The installation menu looks like this:
```
**********************************************************
                    Pactables
**********************************************************
 crebase    :  creation of a new database
**********************************************************
Type the command name or 'x' to quit tabadmin :
```

Description of commands:
- The 'crebase' command starts up the installation process.

Enter the command and follow the instructions given by tabadmin.

Input of the database code (except for the test database)

Enter the database code on 4 characters and its label on 30 characters (alphanumeric uppercase characters).

When installing the first database (test database), the database code is always PTST.

Selection of the Database language code

This code can be:
- en (English)
- fr (French)

Input of the socket number allocated to the listener

The socket number is a 5-digit number between 49152 and 65535.

It must be declared in the /etc/services file (see paragraph 'Update of the "/etc/services" file').

Creation of the Database subdirectories
- $TABDIR/data/PTST/base
- $TABDIR/data/PTST/save
- $TABDIR/data/PTST/script
- $TABDIR/data/PTST/tmp
- $TABDIR/data/PTST/users
- $TABDIR/config/PTST

Installation of the test database

Two test databases are supplied: a French database located in the $TABDIR/system/install/basefra directory and an English database located in the $TABDIR/system/install/baseeng directory.

These test database contain:
- in the base directory, the user files TD, TV and TG,
- in the save directory, the backup file TC,
- in the script directory, the procedures' startup scripts.

At test database installation, these files are copied respectively in the directories:
- $TABDIR/data/"database_name"/base,
- $TABDIR/data/"database_name"/save,
- $TABDIR/data/"database_name"/script.

Update of the journal_tabinstall file

Addition, in the journal_tabinstall file, of information related to the database creation:
- database name,
- creation date.

Delete a database

To delete a database, you must be positioned in the connection directory of the installation user and delete the following directories:

- $TABDIR/data/[database_name]
- $TABDIR/config/[database_name]

To do this, use the command: rm -r [directory_name]

Connection

To allow terminals to connect to Pactables, the listener must be started-up.

You can access the online server via a 3270 emulator.

To access the Pactables database in 3270 mode via the online server, the emulator must be configured accordingly, i.e. you must indicate:

- the IP address of the machine where the on-line server is installed,
- the on-line server port number, chosen at installation time when the database is created.

The code page of the emulator must be valorized according to the database language code:

- code page 1147 for a French database,
- code page 1146 for an English database.

These code pages are set automatically at database creation.

Once the emulator is started up, you must specify a transaction code:

- TABPARM to start up the Pactables transaction of user parameters,
- TABLES to access the specifications Pactables database.

Use tests

These test jobs include the following steps:

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

The test set comprises three tables:

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

- 'ARTICL' with historical account 15/01/1987.

Tests on the user parameters transaction:
- Submit the Paclink Administrator transaction. The user code defined in the test Database is the manager's: user code '********' and password 'SUPER'.
- Use this code to enter the transaction and define other users, then modify the Database manager's user code password.

On-line tests:
- Start up an on-line server; connect it to the workstation.
- Consult all the screens.
- Execute the updates.

Batch tests:
- Execute the PRTA procedure.
- Execute the EXTA procedure.
- Close the on-line server.
- Execute the UPTA procedure.
- Reorganization of the test table:
  - Save TV and TD files (COPY under different names).
  - Execute the reorganization (RETA) which comprises:
    - Reorganization of TV file (Prog. PTA400 and PTA410),
    - Reorganization of TD file (Prog. PTA420),
    - Constitution of the TC backup file (Prog. PTA430),
  - Restore the TV and TD files (RSTA),
    - Execute a printing (PRTA) for verification.
    - Start up the on-line server and a workstation.
    - Do some verification tests after reorganization.

Table generation tests (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.

Chapter 4. Server re-installation

Re-installation

Introduction

You must re-install the system environment of Pactables when you receive a new version with corrections of bugs or new developments.

Generally, only the system file of error messages and the programs are affected by the version.

GENERAL COMMENTS
- The re-installation procedure does not create directories which are assumed to be identical to those created during the initial installation of the version.
- It does not copy the batch procedure ($TABDIR/system/proc directory) so as not to delete possible adaptations to the site, except if the new version cannot run with the old procedures.

  The new procedures are copied into the following directory:

    `$TABDIR/system/proc.Vnn (nn = version number)`
- Operational startup scripts, in the TABDIR/data/[database_name]/script, are not impacted. To get the last version of the scripts, at re-installation, refer to the startup script models in the /system/install/basefra/script or /system/install/baseeng/script directory.

Re-installation procedure start-up

WARNING: The server of the Pactables database must be stopped.

After downloading into the administrator's login directory or mounting the CDROM, go to the following directory:

    `PT350"hardware_code"."version_code"`

As in an installation operation (see sub-chapter 'System installation'), type the command:

    `sh tabinst.Vnnx (nn = version number, x = subversion number)`

followed by a carriage return (Enter or Return key).

Description of steps

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

- consistency check on the environment variables (.profile, .kshrc or .login file),
- display of the re-installation menu,
- choice of COBOL run-time in use (if necessary),
- distribution of the files coming from TABBASE.xxx,
- update of the journal_tabinstall file.

Consistency check of the environment variables

See subchapter 'System installation'.

Display of the re-installation

The re-installation menu is displayed as follows:
```
************************************************************
                    Pactables
************************************************************
 reinst      : version reinstallation
************************************************************
Type the command name or 'x' to quit TABINST.Vnnx :
```

Description of commands:

You perform an overall re-installation via the 'reinst' command.

Enter the command and follow the instructions given by TABINST.Vnnx.

Choice of Cobol run-time in use

See the subchapter 'System installation'.

Update of the journal_tabinstall file

Adding to the journal_tabinstall file information concerning the re-installation:
- re-installation date,
- release number,
- version number.

Chapter 5. The batch procedures

Introduction

The Batch processes associated with Pactables are grouped in procedures. The next chapters present each of these procedures and specify their execution conditions.

For each procedure there is:
- An introduction comprising:
  - the procedure's functionality,
  - the execution conditions,
  - the actions to carry out in case of abnormal execution.
- The description of the user input, processes, and results obtained, as well as advice on use.
- Description of steps:
  - the notations or parameters in use,
  - the list of files in use (temporary and permanent),
  - the return codes for each step.

Classification of procedures

The following procedures are associated with batch processes:
- INTA: Initialization of Pactables files,
- GETT: Table generation,
- UPTA: Table update,
- PRTA: Table printing,
- IMTA: Table import,
- RETA: Table reorganization,
- SVTA: Table backup,
- RSTA: Table restoration,
- LDTA: List of tables descriptions,
- PMTA: User parameters update,
- EXTA: Table extraction,
- TUTA: Table production turnover.

For the Dispatched Table Management (DTM option):
- Table description comparison (CDT1, CDT2),

- Table contents update (CVTA).

For converting from another platform:
- Pactables Database migration (TCTA).

Reminder

Pactables does not include a journal of update transactions.

Abnormal executions

Abends may occur during the execution of a batch program. Input-output errors on the system files or on the Database cause a forced abnormal end with an error code, described in a message displayed on the screen.

When an abend occurs, you must find the error message. This message is displayed in the following manner:

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

This message appears by setting, previously, the "NoDisp" variable to "NO" in Init.vbs.

In most cases, examining the status and type of operation enables you to find the cause of the abnormal execution.

The summary table below lists the most common values for the status and type of operation.

Code	Operation
W	WRITE
RW	REWRITE
RU	READ UPDATE
OP	OPEN
CL	CLOSE
D	DELETE
R	READ
P	START
RN	READ NEXT

Status	Message
21	Sequence error
22	Duplicate key
23	No record found
24	Boundary violation
30	System error
34	Boundary violation (sequential)
92	Logic error (For example, the opening of an already opened file)
93	File still open on line
95	Invalid or Incomplete file

When this message is absent, and the type of ABEND generated directly reports a problem in the VisualAge Pacbase system programs, contact the VisualAge Pacbase support at IBM. KEEP ALL LISTINGS that may be necessary to analyze the problem.

If the error is not an input-output error on a Database file, the following message is displayed:

```
Run Time Error nnn
```

where nnn is the error number.

The Run Time Error 013 is the most frequent. It indicates that the procedure did not find an input file.

The next subchapter contains the list of the most frequent errors. Each Run Time Error is briefly described.

If the Run Time Error is not in the following list or if its associated description is not explicit enough and if the error directly involves the system programs, you must contact the Hot Line and keep all listings which might be useful in solving the problem.

List of ″**Run Time Error**″

This list is a reminder of 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 big
        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 !
```

Management of errors in the procedures

If an error is detected in a step, the following steps are not executed. The name of the erroneous program and, if possible, the type of the detected error, are displayed.

The procedure then displays the message:

```
"Press Return to carry on"
```

You must then stop the procedure, in order to view the error if several procedures follow one another.

(If the NOBVPERR environment variable is set to 'yes', this message is not displayed and you do not have to stop the procedure)

The procedure stops with a return code other than zero. This code can be retrieved via the Return variable right after the command which submits the procedure. This prevents the execution of the next procedures if various procedures are executed in sequence.

Executing procedures

The command files of the procedures are created under the TABDIR/system/proc directory during the installation step.

To run a procedure, you can:

- Directly execute the command file of the batch procedure followed by these parameters:

  ```
  procedure "database_code" -i "user_input_file" +t
  ```

```
                              -u "user_directory" -t "temporary_directory"
```
- or execute the batch procedure via a start-up script:

  This script, written in UNIX shell, sets the environment variables (optionally, the user input) and executes the command file of the procedure.

  An example of operational script is supplied for most of the procedures and for each database created, in the following directory:

  ```
  $TABDIR/data/"database_code"/script
  ```

  In any case, the supplied user inputs must be verified to conform to your environment.

Structure of a procedure

The database manager must sometimes modify the command files of the batch procedures.

For example, if he/she wishes to save the TD and TV files on two different disks or move the TE file, the resulting changes in the command files may be very important.

This is the reason why the procedures are created in such a way that each change in the standard installation is made easily and changes to fit the operating constraints are limited.

The purpose of this subchapter is to analyze the batch procedure so as to explain how it is working and so as to help the user in his/her fitting process.

The parameters

- The Database code (4 characters):

  It is required.
- The complete name of the users input file:

  It is required when the procedure is directly executed.
- Parameter "+t":

  It is optional and is used to prevent the default clearing of temporary files.
- User directory:

  It is optional and it is used to change the user directory default assignment.
- Temporary file:

  It is optional and it is used to change the default assignment of the temporary files directory.

The environment variables

- BVPINPUT :

  This variable contains the user input and is assigned as follows:

```
BVPINPUT=`cat <<eof
1rst user line
2nd  user line
    .
    .
eof`
export BVPINPUT
```

If the : $ ` ' " characters are used, they must be preceded by two \.

This variable has no consequence if the -i parameter is used.

- TABUTI :

  This variable contains the user code, which will be used assign the "users"
  and "tmp" directories. It is set by default with the Pactables user code in
  the user input.

  It is required if the user input is not filled in or if it does not include any
  Pactables user code.

  The assignment process is made as follows:
  ```
  TABUTI="user_code"
  export TABUTI
  ```

- TABBASE :

  This variable contains the database code, which replaces the code entered as
  a parameter to the procedure:

  The assignment process is made as follows:
  ```
  TABBASE="database_code"
  export TABBASE
  ```

- NOBVPPAUSE :

  If it is set to "yes", this variable inhibits any pause during the running of
  the procedure when information messages are displayed.

  The assignment process is made as follows:
  ```
  NOBVPPAUSE="yes"
  export NOBVPPAUSE
  ```

- NOBVPERR :

  If it is set to "yes", this variable inhibits any pause in the running of the
  procedure when error messages are displayed.

  The assignment process is made as follows:
  ```
  NOBVPERR="yes"
  export NOBVPERR
  ```

- "procedure"_INPUT :

  This variable enables you to indicate the full path (directory and name) of
  the file containing the user input.

  The assignment process is made as follows:

```
"procedure"_INPUT="directory/file"
export "procedure"_INPUT
```

• Some environment variables are also used to change the default
  assignment of temporary files and of users files produced (reports or output
  files), either throughout the whole procedure, or only during one step in
  the procedure execution.

  The assignment process is made as follows:

```
"procedure"_"file_code"="directory/file"
export "procedure"_"file_code"
```

or

```
"step"_"file_code"="directory/file"
export "step"_"file_code"
```

Step names and file codes are described in the 'Description of steps' section
for each procedure.

Display and verification of parameters

The execution of a procedure starts with the execution of the command file:

```
. $TABDIR/system/proc/TABINIT.ini
```

This file is created during the installation in the TABDIR/system/proc
directory. It controls the parameters of the procedure.

If it detects an anomaly, TABINIT.ini displays the corresponding error
message and stops the procedure with a return code equal to 20.

If it does not detect any anomaly, the procedure then displays the directories
assignments.

In order for you to view these assignments, at least during installation tests,
the execution can be stopped momentarily with the following message:

```
**********  Check your parameters   *********

Press Control_C to stop the execution

Press Return to carry on
```

If you do not want to stop the execution momentarily, you must set the
NOBVPPAUSE environment variable to 'yes'.

Assignment and codification of files

Each step must be assigned the adequate files.

- THE DATABASE FILES

You assign these files by calling the commands files, created upon installation
in the directory:

```
      $TABDIR/config/"database_name"
```

Example of the assignment of the TD file:
```
    . $TABDIR/config/$1/PAC7TD.ini
```

The main interest in these files is to centralize the assignment of each database file in a single place.

The user who wants to modify the standard location of a file only has to adapt the assignment file.

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

- THE BACKUP FILE

This file is assigned by calling the commands files, created upon installation in the directory:
```
    $TABDIR/config/"database_name"
```

Example of the assignment of the TC file:
```
    . $TABDIR/config/$1/PACSAVTC.ini
```

By default, the TC file is located in the TABDIR/data/$1/save directory.

The name of the backup file used by batch procedures is standardized:
```
    input back-up file (read) = TC
    output back-up file (created by the procedure) = TC.NEW
```

This simplifies the management of this file (see for example the 'Backup files management' section a little further on).

- OUTPUT REPORTS AND FILES

The location of output reports and files is determined by a call to the PACUSERS.ini command file:
```
    . $TABDIR/config/$1/PACUSERS.ini
```

This file is created when a Database is created in the directory:
```
    $TABDIR/config/'database_name' .
```

It contains:
```
# Command file for assignment of TABUSERS environment
# variable
# ( 'users' directory )
```

```
# Description of parameters : $0     = database name
#                           $TABUTI = Pactables user code
TABUSERS=$TABDIR/data/$1/users/$TABUTI
export TABUSERS
```

The use of the -u parameter replaces this default assignment.

When a procedure is executed, a subdirectory named
"procedure_code"_"process_number" is created in the TABUSERS directory.

In the PRTA procedure, the process number is replaced by job number.

The names of the output reports start with the code of the procedure which
outputs them.

More precisely, the reports are coded on nine characters plus an extension
(.txt), in the following manner:
- the first four characters correspond to the procedure code,
- the next two correspond to the last two characters of the file (ID in
  PAC7ID),
- the last three characters correspond to the last three characters of the
  program code (290 in PTA290).

Example : LDTA procedure, PTA290 program
```
    PAC7ID report     --> LDTAID290.txt
For the result files codification, refer to the
'Description of steps' section of each procedure.
```

- TEMPORARY FILES

The location of temporary files is determined by the call to the PACTMP.ini
command file:
```
    . $TABDIR/config/$1/PACTMP.ini
```

This file is created when a database is created in the directory:
```
    $TABDIR/config/'database_name'
```

It contains::
- the assignment of the temporary files directory: TABTMP,
- the disk space required for sorts: SORTFLAG,
- variables specific to each runtime (Microfocus or AcuCobol),
- the directory of sort temporary files: TMPDIR.

The use of the -t parameter replaces this default assignment.

When a procedure is executed, a sub-directory named "procedure_code"_"process_number" is created in the TABTMP directory.

In the PRTA procedure, the process number is replaced by the job number.

The coding of temporary files is explained in the 'Description of steps" section for each procedure.

Advice on use

The objective of this subchapter is to make the person responsible for the database aware of the specifics of the Pactables procedures executed on the UNIX system.

General remarks

1. Each procedure must be passed parameters. All the parameters which may be called in a procedure must be present, even if they are not actually used.
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.

    For a user input directly entered in the script, if the : $ `' " characters are used, they must be preceded by two \.
3. No protection is guaranteed if a BATCH procedure updating the Database system or evolving files is started up while users are interactively updating these same files. One person (the Database manager) must be able to start up the batch procedures which update the Database. He/she therefore must ensure the 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 destroyed at the end of the procedure, except if there was an abend and if a return code other than 0 is sent.
5. The batch procedures must be executed from the UNIX machine.
6. The presence of special characters in the entities code is highly NOT recommended. The EURO character, for example, is source of problems on ACU.

Temporary files management

For each procedure you should consult the corresponding chapter for a detailed description of these files.

In all cases, enough disk space should be freed in the chosen user directory to ensure that the procedure runs smoothly.

Sort temporary files:

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

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

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

If the default directory is too small, the TMPDIR directory assigns another directory for the temporary sort files:

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

Backup files management

All the procedures which create a backup call a command file at the end of an execution without error.

This file is in the $TABDIR/config/"database_name" directory and is called TCBACKUP.ini. it is created when the database is created. It contains:

```
 Rotation script of the journal backup file
. $TABDIR/config/PTST/PACSAVTC.ini
if [ -f "$PACSAVTC" ]
then
   mv -f $PACSAVTC $PACSAVTC'-1'
fi
mv -f $PACSAVTCNEW $PACSAVTC
```

Characteristics of the TCBACKUP file:
*   proceeds by 'mv' to avoid copies of the backup files (these copies may take a long time),
*   guarantees that the TC file is definitely the last backup.

This file does not claim to cover all the operation constraints of all sites. The database manager generally has to adapt it, taking the characteristics above into account.

INTA - Table initialization

INTA - Introduction

This procedure initializes the files containing the descriptions and contents of tables.

Note:

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

INTA - User input

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

INTA - Description of steps

Initialization of files: PTAINI

Code	Physical name	Type	Label
PAC7MD		Input	Input file
PAC7TD	Base dir.: TD	Output	Table-description file
PAC7TV	Base dir.: TV	Output	Table-contents file
PAC7ED	User dir.: INTAEDINI.txt	Report	Initialization review

INTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) INTA BATCH PROCEDURE
# * -------------------------------------------------------
# *    VA PACTABLES
# *
# * -------------------------------------------------------
# *                     - TABLE INITIALIZATION -
# *
# * -------------------------------------------------------
# *
# * INPUT
# * COL 1-36   : INSTALLATION LABEL
# * COL 37     : LANGUAGE VERSION PARAMETER
# *               "E" ENGLISH
# *               "F" FRENCH
# * COL 38     : NOT USED
# * COL 39-50  : FUNCTION KEYS ASSIGNMENTS
# * COL 51-54  : "CCCC" SECURITY SYSTEM CLASS
# * COL 55     : SECURITY SYSTEM TYPE
# *               BLANK : NOT USED
# *               "R"     RACF
# *               "S"     TOP SECRET
# * COL 56-57  : "NN" NUMBER OF LINES PER PRINTOUT PAGE
# * COL 58     : TYPE OF RESOURCE CONTROL
# *             : BLANK : TABLE RESOURCES SECURITY SYSTEM
# *             : "P" :   RESOURCE IN VA PAC
# * COL 59     : LOCK OF USER'S CODE
# *             : BLANK : OTHER USER'S CODE AUTHORIZED
# *             : "N" :   OTHER USER'S CODE UNAUTHORIZED
# *
# * -------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "---------------------------------------------------------"
TABMSG 1004 "INTA"
echo "                         =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "---------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MD=$TABINPUT
export PAC7MD
PAC7ED=`TABENV BVPTAINI PAC7ED $TABUSERS/INTAEDINI.txt`
export PAC7ED
TABMSG 1009 "BVPTAINI"
rtstab BVPTAINI
```

```
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAINI"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

GETT - Table generation

GETT - Introduction

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

Execution condition

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

User input

Result of GETD or GETA extraction.

GETT - Description of steps

Update of table files: PTA250

Code	Physical name	Type	Label
PAC7MD		Input	Transaction file (GETD or GETA output)
PAC7TD	Base dir.: TD	Input Output	Tables descriptions file
PAC7TV	Base dir.: TV	Input Output	Tables contents file
PAC7TK	Tmp dir.: TK	Output	Output file
PAC7ET	User dir.: GETTET250.txt	Report	Input/output errors on files

Printing of descriptions: PTA290

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	Tmp dir.: TK	Input	Print request
PAC7ID	User dir.: GETTID290.txt	Report	Printout of descriptions

GETT - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) GETT BATCH PROCEDURE
# * ------------------------------------------------------
# *    VA PACTABLES
# *
# * ------------------------------------------------------
# *                       - TABLE GENERATION -
# *
# * ------------------------------------------------------
# *
# * INPUT : RESULT OF GETA OR GETD PROCEDURE
# *
# * ------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-------------------------------------------------------------"
TABMSG 1004 "GETT"
echo "                           =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-------------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MD=$TABINPUT
export PAC7MD
PAC7TK=`TABENV BVPTA250 PAC7TK $TABTMP/TK`
export PAC7TK
PAC7ET=`TABENV BVPTA250 PAC7ET $TABUSERS/GETTET250.txt`
export PAC7ET
TABMSG 1009 "BVPTA250"
rtstab BVPTA250
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA250"
```

```
         TABMSG 1025
         TABERR
         TABRMTMP
         exit $RETURN
         ;;
        esac
        # *************************************************************
        . $TABDIR/config/$1/PAC7TD.ini
        PAC7TE=`TABENV BVPTA290 PAC7TE $TABTMP/TK`
        export PAC7TE
        PAC7ID=`TABENV BVPTA290 PAC7ID $TABUSERS/GETTID290.txt`
        export PAC7ID
        TABMSG 1009 "BVPTA290"
        rtstab BVPTA290
        RETURN=$?
        case $RETURN in
        0)
         ;;
        *)
         TABMSG 1012 "BVPTA290"
         TABMSG 1025
         TABERR
         TABRMTMP
         exit $RETURN
         ;;
        esac
        # *************************************************************
        TABMSG 1010
        TABRMTMP
        exit $RETURN
```

UPTA - Table update

UPTA - Introduction

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

Execution condition

None.

Important note

An alternative version of the update program, PTA302, is available with Pactables 2.0 and higher.

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

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

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

UPTA - User input

- One '*'-type line per user:

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

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

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

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

Pos.	Len.	Value	Meaning
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	1	'V'	Line code
3	1		Continuation line
		' '	First data line
		'-'	Item data continuation

Pos.	Len.	Value	Meaning
4	77		Table data separated by the delimiter indicated on the 'A'-type line

UPTA - Description of steps

Table update: PTA300

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TG	Base dir.: TG	Input	User parameters file
PAC7TV	Base dir.: TV	Input Output	Tables contents file
PAC7MS		Input	Update transactions
PAC7DE	Tmp dir.: DE	Output	Print requests
PAC7ET	User dir.: UPTAET300.txt	Report	Transaction review
PAC7MT	Tmp dir.: MT		Prepared transactions

Formatting for printing: PTA350

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7DE	Tmp dir.: DE	Input	Print request
PAC7ET	Tmp dir.: ET	Output	Print file
PAC7EX	User dir.: UPTAEX350.txt	Report	Statistics on printing

Printing of tables: PTA360

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7ET	Tmp dir.: ET	Input	Print file
PAC7EY	User dir.: UPTAEY360.txt	Report	Printing of tables

UPTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) UPTA BATCH PROCEDURE
# * ---------------------------------------------------------
# *    VA PACTABLES
# *
# * ---------------------------------------------------------
# *                    - TABLE UPDATE -
# *
# * ---------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE
# * COL 11-18  : PASSWORD
# *
# * --> ONE "A" TYPE LINE PER TABLE TO UPDATE
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# * COL 17     : NOT USED
# * COL 18     : SUB-SYSTEM NUMBER
# *            : " "     : NO SUB-SYSTEM SPECIFIED
# *            : 1 TO 0 : SUB-SYSTEM NUMBER
# * COL 19     : DATA DELIMITER
# *            : " " : CONSIDER AS "/" BY DEFAULT
# * --> ONE "V" TYPE LINE TO UPDATE TABLE DATA
# * COL 1      : ACTION CODE
# *            : "C" :   CREATION
# *            : "M" :   MODIFICATION
# *            : "D" :   DELETION
# * COL 2      : "V" :   LINE CODE
# * COL 3      : CONTINUATION LINE
# *            : " " : FIRST DATA LINE
# *            : "-" : ITEM DATA CONTINUATION
# * COL 4-80   : TABLE DATA SEPARATED BY THE DELIMITER
# *            : INDICATED ON THE "A" TYPE LINE
# *
# * ---------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-------------------------------------------------------------"
TABMSG 1004 "UPTA"
echo "                        =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-------------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
```

```
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MS=$TABINPUT
export PAC7MS
PAC7DE=`TABENV BVPTA300 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7MT=`TABENV BVPTA300 PAC7MT $TABTMP/MT`
export PAC7MT
PAC7ET=`TABENV BVPTA300 PAC7ET $TABUSERS/UPTAET300.txt`
export PAC7ET
TABMSG 1009 "BVPTA300"
rtstab BVPTA300
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA300"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7DE=`TABENV BVPTA350 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA350 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EX=`TABENV BVPTA350 PAC7EX $TABUSERS/UPTAEX350.txt`
export PAC7EX
TABMSG 1009 "BVPTA350"
rtstab BVPTA350
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA350"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
. $TABDIR/config/$1/PAC7TD.ini
PAC7ET=`TABENV BVPTA360 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EY=`TABENV BVPTA360 PAC7EY $TABUSERS/UPTAEY360.txt`
export PAC7EY
```

```
TABMSG 1009 "BVPTA360"
rtstab BVPTA360
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA360"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

PRTA - Table printing

PRTA - Introduction

This procedure performs a batch printing of tables.

Execution condition

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

Note

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

PRTA - User input

- One '*'-type line per user:

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

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

Pos.	Len.	Value	Meaning
1	1		Action code
		'E'	Table printing

Pos.	Len.	Value	Meaning
		'H'	List of historical accounts
		'L'	List of the tables
		'S'	List of sub-schemas and sub-systems
		'X'	Table contents with historical accounts
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Historical account date or date of the reference description (if transaction code = 'X')
17	1		Sub-schema selection
		blank	No sub-schema selection
		1 to 0	Selected sub-schema number
18	1		Sub-system selection
		blank	No sub-system selection
		1 to 0	Selected sub-system number
19	1		Print option of the key's data elements
		blank	Printing of concatenated data elements
		'O'	Printing of separated data elements

PRTA - Description of steps

Extraction of tables for printing: PTA320

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TV	Base dir.: TV	Input	Table contents File
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7CA		Input	Update transactions
PAC7DE	Tmp dir.: DE	Output	Print requests
PAC7XE	User dir.: PRTAXE320.txt	Report	Transaction review

Preparation for printing: PTA350

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TV	Base dir.: TV	Input	Tables contents file

Code	Physical name	Type	Label
PAC7DE	Tmp dir.: DE	Input	Print requests
PAC7ET	Tmp dir.: ET	Output	Print file
PAC7EX	User dir.: PRTAEX350.txt	Report	Statistics on printing

Printing of tables: PTA360

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7ET	Tmp dir.: ET	Input	Print file
PAC7EY	User dir.: PRTAEY360.txt	Report	Printing of tables

PRTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) PRTA BATCH PROCEDURE
# * ------------------------------------------------------
# *    VA PACTABLES
# *
# * ------------------------------------------------------
# *                    - TABLE PRINT -
# *
# * ------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2       : "*" : LINE CODE
# * COL 3-10    : USER CODE
# * COL 11-18   : PASSWORD
# *
# * --> ONE "A" TYPE LINE PER TABLE TO BE PRINTED
# * COL 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
# * COL 2       : "A" : LINE CODE
# * COL 3-8     : TABLE NUMBER (TTTTTT)
# * COL 9-16    : HISTORICAL ACCOUNT DATE OR DATE OF THE
# *             : REFERENCE DESCRIPTION
# * COL 17      : SUB-SCHEMA SELECTION
# *             : BLANK  : NO SUB-SCHEMA SELECTION
# *             : 1 TO 0 : SELECTED SUB-SCHEMA NUMBER
# * COL 18      : SUB-SYSTEM SELECTION
# *             : BLANK  : NO SUB-SYSTEM SELECTION
# *             : 1 TO 0 : SELECTED SUB-SYSTEM NUMBER
# * COL 19      : PRINT OPTION OF THE KEY'S DATA ELEMENTS
```

```
# *              : BLANK : CONCATENED DATA ELEMENTS
# *              : "O"   : SEPARATED DATA ELEMENTS
# *
# * -------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "----------------------------------------------------------"
TABMSG 1004 "PRTA"
echo "                              =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7CA=$TABINPUT
export PAC7CA
PAC7DE=`TABENV BVPTA320 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7XE=`TABENV BVPTA320 PAC7XE $TABUSERS/PRTAXE320.txt`
export PAC7XE
TABMSG 1009 "BVPTA320"
rtstab BVPTA320
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA320"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7DE=`TABENV BVPTA350 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA350 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EX=`TABENV BVPTA350 PAC7EX $TABUSERS/PRTAEX350.txt`
export PAC7EX
TABMSG 1009 "BVPTA350"
rtstab BVPTA350
RETURN=$?
```

```
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA350"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
. $TABDIR/config/$1/PAC7TD.ini
PAC7ET=`TABENV BVPTA360 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EY=`TABENV BVPTA360 PAC7EY $TABUSERS/PRTAEY360.txt`
export PAC7EY
TABMSG 1009 "BVPTA360"
rtstab BVPTA360
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA360"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

IMTA - Table import

IMTA - Introduction

This procedure imports external tables into the existing Pactables files.

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

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

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

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

Execution condition

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

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

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

Restriction

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

Important note

An alternative version of the update program, PTA312, is shipped with Pactables 2.0 and higher releases.

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

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

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

IMTA - User input

- One '*'-type line per user:

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

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

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

IMTA - Description of steps

Table check and update: PTA310

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir. : TE	Input	Error message file
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7TV	Base dir.: TV	Input Output	Tables contents files
PAC7MV		Input	Request transactions
PAC7NK	User dir.: TABF.txt	Input	External table file
PAC7DE	Tmp dir.: DE	Output	Print requests
PAC7ET	User dir.: IMTAET310.txt	Report	Execution report

Formatting of printout: PTA350

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7DE	Tmp dir.: DE	Input	Print requests
PAC7ET	Tmp dir.: ET	Output	Print file
PAC7EX	User dir.: IMTAEX350.txt	Report	Printing statistics

Printing: PTA360

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7ET	Tmp dir.: ET	Input	Print file
PAC7EY	User dir.: IMTAEY360.txt	Report	Table printout

IMTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) IMTA BATCH PROCEDURE
# * ----------------------------------------------------------
# *    VA PACTABLES
# *
# * ----------------------------------------------------------
# *                        - TABLE IMPORT -
# *
# * ----------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE
# * COL 11-18  : PASSWORD
# *
# * --> ONE "A" TYPE LINE PER TABLE TO BE IMPORTED
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER TO BE IMPORTED (TTTTTT)
# * COL 9-16   : TABLE DATE (OPTIONAL) (DDMMCCYY)
# *
# * ----------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "--------------------------------------------------------"
TABMSG 1004 "IMTA"
echo "                        =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "--------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7NK=`TABENV BVPTA310 PAC7NK \`dirname $TABUSERS\`/TABF.txt`
export PAC7NK
PAC7DE=`TABENV BVPTA310 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA310 PAC7ET $TABUSERS/IMTAET310.txt`
export PAC7ET
TABMSG 1009 "BVPTA310"
rtstab BVPTA310
RETURN=$?
case $RETURN in
0)
```

```
 ;;
*)
 TABMSG 1012 "BVPTA310"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TV.ini
PAC7DE=`TABENV BVPTA350 PAC7DE $TABTMP/DE`
export PAC7DE
PAC7ET=`TABENV BVPTA350 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EX=`TABENV BVPTA350 PAC7EX $TABUSERS/IMTAEX350.txt`
export PAC7EX
TABMSG 1009 "BVPTA350"
rtstab BVPTA350
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA350"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
. $TABDIR/config/$1/PAC7TD.ini
PAC7ET=`TABENV BVPTA360 PAC7ET $TABTMP/ET`
export PAC7ET
PAC7EY=`TABENV BVPTA360 PAC7EY $TABUSERS/IMTAEY360.txt`
export PAC7EY
TABMSG 1009 "BVPTA360"
rtstab BVPTA360
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA360"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
```

```
# **********************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

RETA - Table reorganization

RETA - Introduction

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

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

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

Execution condition

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

RETA - User input

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

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

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

Pos.	Len.	Value	Meaning
1	1		Action code
		'S'	Historical account to purge
		'G'	Historical account to keep
2	1	'A'	Line code
3	6	ttttt	Table number
9	8	DDMMCCYY	Historical account date
19	1	' '	Option

Pos.	Len.	Value	Meaning
			- when the action code is equal to 'G', the historical account whose date is equal to the date specified is kept. If there is no date, all historical accounts are kept.
			- When the action code is equal to 'S', the historical account whose date is equal to the date specified is purged.
		'<'	- When the action code is equal to 'G', the historical accounts whose dates are strictly earlier than the date specified are kept.
			- When the action code is equal to 'S', all historical accounts whose dates are strictly earlier than the date specified are purged.
		'>'	- When the action code is equal to 'G', all historical accounts whose dates are later than or equal to the date specified are kept.
			- When the action code is equal to 'S', all historical accounts whose dates are later than or equal to the date specified are purged.

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

For more details, see the Pactables Reference Manual.

RETA - Description of steps

Reorganization of table contents: PTA400

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error-message file
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7DR		Input	Reorganization requests
PAC7TX	Tmp dir.: TX	Output	Reorganized contents file
PAC7DE	User dir.: MVRETA.txt	Output	Reorganized table list file
PAC7IR	User dir.: RETAIR400.txt	Report	Transaction report

Note:

The PAC7DE file (reorganized table list file) whose description contains print requests, may be kept. Once the reorganization is complete, it can be used as

input for the PRTA procedure applied to the reorganized files, thus enabling the printing of all the tables that were kept, in order to check the correct execution of the reorganization.

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

Validation of table contents: PTA410

Code	Physical name	Type	Label
PAC7MB		Input	Reorganization requests
PAC7TX	Tmp dir.: TX	Input	Reorganized contents file
PAC7TW	Tmp dir.: TW	Output	Validated contents file

Reorganization of table-descriptions: PTA420

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7DE	User dir.: MVRETA.txt	Input	Reorganized table list file
PAC7TS	Tmp dir.: TS	Output	Reorganized tables descriptions file
PAC7ML	User dir.: MWRETA.txt	Output	Tables descriptions print request

Note

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

Building of backup file: PTA430

Code	Physical name	Type	Label
PAC7TW	Tmp dir.: TW	Input	Validated contents file
PAC7TS	Tmp dir.: TS	Input	Reorganized descriptions file
PAC7TC	Save dir.: TC	Output	Backup file resulting from reorganization

TG file backup: PTASVG

Code	Physical name	Type	Label
PAC7TG	Base dir.: TG	Input	User-parameter file
PAC7TC	Save dir.: TC	Output	Table backup

RETA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) RETA BATCH PROCEDURE
# * ------------------------------------------------------
# *    VA PACTABLES
# *
# * ------------------------------------------------------
# *                    - TABLE REORGANIZATION -
# *
# * ------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE IDENTIFYING PACTABLES MANAGER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : PACTABLES MANAGER CODE
# * COL 11-18  : PACTABLES MANAGER PASSWORD
# *
# * --> ONE "A" TYPE LINE PER HISTORICAL ACCOUNT TO KEEP
# *      OR DELETE
# * COL 1      : ACTION CODE
# *            : "S" : HISTORICAL ACCOUNT TO PURGE
# *            : "G" : HISTORICAL ACCOUNT TO KEEP
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# * COL 19     : OPTION
# *            : " "  : SEE DOCUMENTATION FOR MORE
# *            : "X"  : EXPLANATIONS
# *            : "X"
# *
# * ------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----------------------------------------------------------"
TABMSG 1004 "RETA"
echo "                          ============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
```

```
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7DR=$TABINPUT
export PAC7DR
PAC7DE=`TABENV BVPTA400 PAC7DE $TABUSERS/MVRETA.txt`
export PAC7DE
PAC7TX=`TABENV BVPTA400 PAC7TX $TABTMP/TX`
export PAC7TX
PAC7IR=`TABENV BVPTA400 PAC7IR $TABUSERS/RETAIR400.txt`
export PAC7IR
TABMSG 1009 "BVPTA400"
rtstab BVPTA400
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA400"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ****************************************************************
PAC7TX=`TABENV BVPTA410 PAC7TX $TABTMP/TX`
export PAC7TX
PAC7TW=`TABENV BVPTA410 PAC7TW $TABTMP/TW`
export PAC7TW
PAC7MB=$TABINPUT
export PAC7MB
TABMSG 1009 "BVPTA410"
rtstab BVPTA410
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA410"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ****************************************************************
. $TABDIR/config/$1/PAC7TD.ini
PAC7TS=`TABENV BVPTA420 PAC7TS $TABTMP/TS`
export PAC7TS
PAC7DE=`TABENV BVPTA420 PAC7DE $TABUSERS/MVRETA.txt`
export PAC7DE
PAC7ML=`TABENV BVPTA420 PAC7ML $TABUSERS/MWRETA.txt`
export PAC7ML
```

```
TABMSG 1009 "BVPTA420"
rtstab BVPTA420
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA420"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
PAC7TW=`TABENV BVPTA430 PAC7TW $TABTMP/TW`
export PAC7TW
PAC7TS=`TABENV BVPTA430 PAC7TS $TABTMP/TS`
export PAC7TS
TABMSG 1009 "BVPTA430"
rtstab BVPTA430
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA430"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
. $TABDIR/config/$1/PAC7TG.ini
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVG"
rtstab BVPTASVG
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTASVG"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
```

```
# ************************************************************
TABMSG 1010
echo ""
echo "Calling the file TCBACKUP.CMD"
. $TABDIR/config/$1/TCBACKUP.ini
TABRMTMP
exit $RETURN
```

SVTA - Backup

SVTA - Introduction

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

Execution condition

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

User input

None.

SVTA - Description of steps

TD backup: PTASVT

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TC	Save dir.: TC	Output	Table backup

TV backup: PTASVV

Code	Physical name	Type	Label
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7TC	Save dir.: TC.NEW	Output	Table backup

TG backup: PTASVG

Code	Physical name	Type	Label
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7TC	Save dir.: TC.NEW	Output	Table backup

SVTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) SVTA BATCH PROCEDURE
# * ---------------------------------------------------------
# *   VA PACTABLES
# *
# * ---------------------------------------------------------
# *                       - TABLES BACKUP -
# *
# * ---------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "----------------------------------------------------------"
TABMSG 1004 "SVTA"
echo "                          =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "----------------------------------------------------------"
echo ""
TABPAUSE
STATUS=`tabtp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != " Inactive" -a "$STATUS" != "" ]
then
   TABMSG 1012 "SVTA"
   TABMSG 1037 $1
   TABERR
   TABRMTMP
   exit $RETURN
fi
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVT"
rtstab BVPTASVT
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTASVT"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
. $TABDIR/config/$1/PAC7TV.ini
PAC7TC=$PACSAVTCNEW
```

```
export PAC7TC
TABMSG 1009 "BVPTASVV"
rtstab BVPTASVV
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTASVV"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
. $TABDIR/config/$1/PAC7TG.ini
PAC7TC=$PACSAVTCNEW
export PAC7TC
TABMSG 1009 "BVPTASVG"
rtstab BVPTASVG
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTASVG"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
TABMSG 1010
echo ""
echo "Calling the file TCBACKUP.ini"
. $TABDIR/config/$1/TCBACKUP.ini
TABRMTMP
exit $RETURN
```

TCTA - Pactables transfer from another platform

TCTA - Introduction

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

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

If the 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. But carefully read the note below.

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.

TCTA - Description of steps

TC backup split: PTATC1

Code	Physical name	Type	Label
PAC7TC	Save dir.: TC	Input	Input backup file
PAC7SD	Tmp dir.: SD	Output	Tables descriptions sequential image
PAC7SV	Tmp dir.: SV	Output	Tables contents sequential image

Code	Physical name	Type	Label
PAC7SG	Tmp dir.: SG	Output	Parameter sequential image

Table-description sort: PTATCD

Code	Physical name	Type	Label
PAC7SD	Tmp dir.: DS	Input	Tables descriptions sequential image
PAC7AD	Tmp dir.: AD	Output	Sorted table descriptions

Table-contents sort: PTATCV

Code	Physical name	Type	Label
PAC7SV	Tmp dir.: SV	Input	Sequential image of tables contents
PAC7AV	Tmp dir.: AV	Output	Sorted tables contents

User-parameter sort: PTATCG

Code	Physical name	Type	Label
PAC7SG	Tmp dir.: SG	Input	Sequential image of parameters
PAC7AG	Tmp dir.: AG	Output	Sorted user parameters

Reconstitution of the TC backup: PTATC2

Code	Physical name	Type	Label
PAC7AD	Tmp dir.: AD	Input	Tables descriptions sequential image
PAC7AV	Tmp dir.: AV	Input	Sequential image of contents
PAC7AG	Tmp dir.: AG	Input	Sequential image of parameters
PAC7TC	Save dir.: TC.NEW	Output	TC backup in EBCDIC format

TCTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) TCTA BATCH PROCEDURE
# * -------------------------------------------------------
# *    VA PACTABLES
# *
# * -------------------------------------------------------
# *        - TABLES TRANSFER FROM ANOTHER PLATFORM -
# *
# * -------------------------------------------------------
# Parameter control
```

```
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "----------------------------------------------------------"
TABMSG 1004 "TCTA"
echo "                              =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
. $TABDIR/config/$1/PACSAVTC.ini
TABMSG 1044 $PACSAVTC.INI
TABMSG 1045 $PACSAVTC
echo "----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTC.INI
export PAC7TC
PAC7SD=`TABENV BVPTATC1 PAC7SD $TABTMP/SD`
export PAC7SD
PAC7SG=`TABENV BVPTATC1 PAC7SG $TABTMP/SG`
export PAC7SG
PAC7SV=`TABENV BVPTATC1 PAC7SV $TABTMP/SV`
export PAC7SV
TABMSG 1009 "BVPTATC1"
rtstab BVPTATC1
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTATC1"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
PAC7SD=`TABENV BVPTATCD PAC7SD $TABTMP/SD`
export PAC7SD
PAC7AD=`TABENV BVPTATCD PAC7AD $TABTMP/AD`
export PAC7AD
TABMSG 1009 "BVPTATCD"
rtstab BVPTATCD
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTATCD"
 TABMSG 1025
 TABERR
```

```
 TABRMTMP
 exit $RETURN
 ;;
esac
# ***************************************************************
PAC7SV=`TABENV BVPTATCV PAC7SV $TABTMP/SV`
export PAC7SV
PAC7AV=`TABENV BVPTATCV PAC7AV $TABTMP/AV`
export PAC7AV
TABMSG 1009 "BVPTATCV"
rtstab BVPTATCV
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTATCV"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ***************************************************************
PAC7SG=`TABENV BVPTATCG PAC7SG $TABTMP/SG`
export PAC7SG
PAC7AG=`TABENV BVPTATCG PAC7AG $TABTMP/AG`
export PAC7AG
TABMSG 1009 "BVPTATCG"
rtstab BVPTATCG
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTATCG"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ***************************************************************
PAC7TC=$PACSAVTCNEW
export PAC7TC
PAC7AD=`TABENV BVPTATC2 PAC7AD $TABTMP/AD`
export PAC7AD
PAC7AG=`TABENV BVPTATC2 PAC7AG $TABTMP/AG`
export PAC7AG
PAC7AV=`TABENV BVPTATC2 PAC7AV $TABTMP/AV`
export PAC7AV
TABMSG 1009 "BVPTATC2"
rtstab BVPTATC2
RETURN=$?
case $RETURN in
```

```
0)
 ;;
*)
 TABMSG 1012 "BVPTATC2"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
TABMSG 1010
echo ""
echo "Calling the file TCBACKUP.ini"
. $TABDIR/config/$1/TCBACKUP.ini
TABRMTMP
exit $RETURN
```

RSTA - Restoration

RSTA - Introduction

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

Execution condition

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

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

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

Abnormal execution

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

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

User input

None.

RSTA - Description of steps

Restoration of TD: PTARST

Code	Physical name	Type	Label
PAC7TC	Save dir.: TC	Input	Table backup
PAC7TD	Base dir.: TD	Output	Tables descriptions file

Restoration of TV: PTARSV

Code	Physical name	Type	Label
PAC7TC	Save dir.: TC	Input	Table backup
PAC7TV	Base dir.: TV	Output	Tables contents file

Restoration of TG: PTARSG

Code	Physical name	Type	Label
PAC7TC	Save dir.: TC	Input	Table backup
PAC7TG	Base dir.: TG	Output	User parameter file

RSTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) RSTA BATCH PROCEDURE
# * ------------------------------------------------------
# *    VA PACTABLES
# *
# * ------------------------------------------------------
# *              - TABLES RESTORATION -
# *
# * ------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-------------------------------------------------------"
TABMSG 1004 "RSTA"
echo "                        =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-------------------------------------------------------"
echo ""
TABPAUSE
STATUS=`tabtp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != " Inactive" -a "$STATUS" != "" ]
then
```

```
      TABMSG 1012 "RSTA"
      TABMSG 1037 $1
      TABERR
      TABRMTMP
      exit $RETURN
fi
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PACSAVTC.ini
PAC7TC=$PACSAVTC
export PAC7TC
TABMSG 1009 "BVPTARST"
rtstab BVPTARST
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTARST"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
. $TABDIR/config/$1/PAC7TV.ini
PAC7TC=$PACSAVTC
export PAC7TC
TABMSG 1009 "BVPTARSV"
rtstab BVPTARSV
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTARSV"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
. $TABDIR/config/$1/PAC7TG.ini
PAC7TC=$PACSAVTC
export PAC7TC
TABMSG 1009 "BVPTARSG"
rtstab BVPTARSG
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTARSG"
```

```
      TABMSG 1025
      TABERR
      TABRMTMP
      exit $RETURN
      ;;
    esac
    # **********************************************************
    TABMSG 1010
    TABRMTMP
    exit $RETURN
```

LDTA - List of table descriptions

LDTA - Introduction

This procedure prints descriptions of tables.

Execution condition

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

LDTA - User input

A 'Z'-type line per print request:

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

Note

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

LDTA - Description of steps

Printing of tables descriptions: PTA290

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE		Input	Print request
PAC7ID	User dir.: LDTAID290.txt	Report	Tables descriptions printout

LDTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) LDTA BATCH PROCEDURE
# * --------------------------------------------------------
# *    VA PACTABLES
# *
# * --------------------------------------------------------
# *              - LIST OF TABLE DESCRIPTIONS -
# *
# * -------------------------------------------------------
# *
# * INPUT
# * --> ONE "Z" TYPE LINE PER PRINT REQUEST
# * COL 2     : "Z" : LINE CODE
# * COL 5-8   : PRINT REQUEST
# *             : "TLS" : LIST OF TABLE DESCRIPTIONS
# *             : "TDS" : TABE DESCRIPTION
# * COL 9-14  : TABLE NUMBER (TTTTTT)
# * COL 23-30 : HISTORICAL ACCOUNT DATE (MMDDCCYY)
# *
# * -------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "----------------------------------------------------------"
TABMSG 1004 "LDTA"
echo "                        ============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
PAC7TE=$TABINPUT
export PAC7TE
PAC7ID=`TABENV BVPTA290 PAC7ID $TABUSERS/LDTAID290.txt`
export PAC7ID
TABMSG 1009 "BVPTA290"
rtstab BVPTA290
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA290"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
```

```
# ************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

PMTA - Parameter update

PMTA - Introduction

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

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

Execution condition

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

PMTA - User input

- 'TA'-line: user parameter updating:

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

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

Pos.	Len.	Value	Meaning
1	1		Action code

Pos.	Len.	Value	Meaning
		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 the last Administrator code is not authorized.

PMTA - Description of steps

User parameters update: PTA100

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TG	Base dir.: TG	Input Output	User parameter file
PAC7MV		Input	Extraction requests
PAC7NU	Tmp dir.: NU	Output	Parameter printing requests
PAC7ET	User dir.: PMTAET100.txt	Report	Printing of descriptions

Printing of user parameters: PTA120

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7NU	Tmp dir.: NU	Input	Print requests
PAC7ET	User dir.: PMTAET120.txt	Report	Printing of user parameters

PMTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) PMTA BATCH PROCEDURE
# * ------------------------------------------------------
# *    VA PACTABLES
# *
# * ------------------------------------------------------
# *            - USER PARAMETER UPDATE -
# *
# * ------------------------------------------------------
# *
# * INPUT
# * --> ONE "TA" LINE : USER PARAMETER UPDATING
# * COL 1      : ACTION CODE
# *             : " " : CREATION OR MODIFICATION
# *             : "C" : CREATION
# *             : "M" : MODIFICATION
# *             : "D" : DELETION
# * COL 2-9    : USER CODE (UUUUUUUU)
```

```
# * COL 10-11  : "TA" : LINE CODE
# * COL 12-19  : PASSWORD (PPPPPPPP)
# * COL 20     : GENERAL ACCESS AUHTORIZATION
# *            : "0" : NO GENERAL ACCESS AUTHORIZATION
# *            : "1" : READ ONLY ACCESS AUTHORIZATION
# *            : "2" : READ WRITE AUTHORIZATION ON TABLES
# *            : "3" : READ WRITE AUTHORIZATION ON USER
# *            :       CODE
# *
# * --> ONE "TC" LINE : ACCESS AUTHORIZATIONS PER TABLE
# * COL 1      : ACTION CODE
# *            : " " : CREATION OR MODIFICATION
# *            : "C" : CREATION
# *            : "M" : MODIFICATION
# *            : "D" : DELETION
# * COL 2-9    : USER CODE (UUUUUUUU)
# * COL 10-11  : "TC" : LINE CODE
# * COL 12-17  : TABLE CODE (TTTTTT)
# * COL 18-20  : LINE NUMBER (NNN)
# * COL 21-80  : ACCESS AUTHORIZATIONS : 20 ACCESS
# *            : AUTHORIZATIONS MAY BE ENTERED IN THIS
# *            : FIELD, WITH, FOR EACH AUTHORIZATION, THE
# *            : SUB-SCHEMA NUMBER (N), THE SUB-SYSTEM
# *            : NUMBER (N), THE AUTHORIZATION (0,1,2).
# *            : "*" IS REQUIRED FOR ALL SUB-SCHEMAS AND
# *            : SUB-SYSTEMS.
# *
# * --> ONE "TJ" LINE : CONTROL CARDS
# * COL 1      : ACTION CODE
# *            : " " : CREATION OR MODIFICATION
# *            : "C" : CREATION
# *            : "M" : MODIFICATION
# *            : "D" : DELETION
# * COL 2-9    : USER CODE (UUUUUUUU)
# * COL 10-11  : "TJ" : LINE CODE
# * COL 12-17  : TABLE CODE (TTTTTT)
# * COL 18-20  : SCRIPT LINE NUMBER
# *            :LT600000 : CONTROL CARD IN FRONT OF PROGRAM
# *            :GT599999 : CONTROL CARD IN BACK OF PROGRAM
# * COL 21-80  : CONTENT OF THE SCRIPT LINE
# *
# * -------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "--------------------------------------------------------"
TABMSG 1004 "PMTA"
echo "                        =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "--------------------------------------------------------"
echo ""
```

```
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7NU=`TABENV BVPTA100 PAC7NU $TABTMP/NU`
export PAC7NU
PAC7ET=`TABENV BVPTA100 PAC7ET $TABUSERS/PMTAET100.txt`
export PAC7ET
TABMSG 1009 "BVPTA100"
rtstab BVPTA100
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA100"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7NU=`TABENV BVPTA120 PAC7NU $TABTMP/NU`
export PAC7NU
PAC7ET=`TABENV BVPTA120 PAC7ET $TABUSERS/PMTAET120.txt`
export PAC7ET
TABMSG 1009 "BVPTA120"
rtstab BVPTA120
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA120"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

EXTA - Table extraction

EXTA - Introduction

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

Execution condition

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

EXTA - User input

- One '*'-type line per user:

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

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

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

EXTA - Description of steps

Extraction of table data: PTA150

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7MV		Input	Extraction requests

Code	Physical name	Type	Label
PAC7EX	Tmp dir.: EX	Output	Extracted transactions
PAC7ET	User dir.: EXTAET150.txt	Report	Transaction review

Printing of extracted transactions: PTA160

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7EX	Tmp dir.: EX	Input	Extracted transactions
PAC7NU	User dir.: NUTAB.txt	Output	Extracted transactions
PAC7ET	User dir.: EXTAET160.txt	Report	Printing of extracted data

Return codes:

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

EXTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) EXTA BATCH PROCEDURE
# * --------------------------------------------------------
# *    VA PACTABLES
# *
# * --------------------------------------------------------
# *                - TABLE EXTRACTION  -
# *
# * ------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE PER TABLE TO EXTRACT
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# * COL 17     : NOT USED
# * COL 18     : SUB-SYSTEM SELECTION
# *            : BLANK : NO SUB-SYSTEM SELECTION
# *            : 1 TO 0 : NUMBER OF SELECTED SUB SYSTEM
# * COL 19     : DATA DELIMITEUR
# *            : BLANK : "/" BY DEFAULT
# *
# * ------------------------------------------------------
```

```
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "----------------------------------------------------------"
TABMSG 1004 "EXTA"
echo "                              =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7EX=`TABENV BVPTA150 PAC7EX $TABTMP/EX`
export PAC7EX
PAC7ET=`TABENV BVPTA150 PAC7ET $TABUSERS/EXTAET150.txt`
export PAC7ET
TABMSG 1009 "BVPTA150"
rtstab BVPTA150
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA150"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
. $TABDIR/config/$1/PAC7TD.ini
PAC7EX=`TABENV BVPTA160 PAC7EX $TABTMP/EX`
export PAC7EX
PAC7NU=`TABENV BVPTA160 PAC7NU \`dirname $TABUSERS\`/NUTAB.txt`
export PAC7NU
PAC7ET=`TABENV BVPTA160 PAC7ET $TABUSERS/EXTAET160.txt`
export PAC7ET
TABMSG 1009 "BVPTA160"
rtstab BVPTA160
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTA160"
```

```
         TABMSG 1025
         TABERR
         TABRMTMP
         exit $RETURN
         ;;
        esac
        # *********************************************************
        TABMSG 1010
        TABRMTMP
        exit $RETURN
```

TUTA - Direct consultation of tables

TUTA - Introduction

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

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

Execution condition

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

The TUTA procedure defines both files in the second step.

TUTA - User input

- One '*'-type line :

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

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

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

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

TUTA - Description of steps

Direct consultation of tables: PTAU80

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7MX		Input	Request transactions
PAC7AD	Base dir.: AD	Output	Tables descriptions file
PAC7AV	Base dir.: AV	Output	Tables contents file
PAC7ET	User dir.: TUTAETU80.txt	Report	Transaction report

TUTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) TUTA BATCH PROCEDURE
# * -------------------------------------------------------
# *    VA PACTABLES
# *
# * -------------------------------------------------------
# *           - DIRECT CONSULTATION OF TABLES -
# *
# * -------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2       : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
# *
# * -------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----------------------------------------------------------"
TABMSG 1004 "TUTA"
echo "                        ============="
TABMSG 1047 "$TABBASE"
```

```
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MX=$TABINPUT
export PAC7MX
. $TABDIR/config/$1/PAC7AD.ini
. $TABDIR/config/$1/PAC7AV.ini
PAC7ET=`TABENV BVPTAU80 PAC7ET $TABUSERS/TUTAETU80.txt`
export PAC7ET
TABMSG 1009 "BVPTAU80"
rtstab BVPTAU80
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAU80"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

Dispatched table management (DTM option)

CDT1-CDT2 - Table descriptions comparison

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

Table description comparison

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

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

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

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

CDT1 - User input

- One '*'-type line per user:

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

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

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

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

CDT1 - Description of steps

Check of transactions: PTAD05

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions Master file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TG	Base dir.: TG	Input	User parameter file
PAC7MV		Input	Comparison request transactions
PAC7MX	Tmp dir.: MX	Output	Validated comparison request transactions
PAC7ET	User dir.: CDT1ETD05.txt	Report	Transaction report

Table-description comparison and extraction: PTAD10

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table description 'Master' file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TS	User dir.: TD	Input	Table description 'Slave' file
PAC7MX	Tmp dir.: MX	Input	Validated transactions
PAC7TX	User dir.: MBCDT2.txt	Output	Comparison result to be used as input of the CDT2 procedure
PAC7ET	User dir.: CDT1ETD10.txt	Report	Extraction printout

CDT1 - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) CDT1 BATCH PROCEDURE
# * ---------------------------------------------------------
# *    VA PACTABLES
# *
# * ---------------------------------------------------------
# *         - DISPATCHED TABLE MANAGEMENT  -
# *         - TABLE DESCRIPTION COMPARISON  -
# * ---------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# *
# * ---------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "---------------------------------------------------------"
TABMSG 1004 "CDT1"
echo "                         =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "---------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
```

```
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7MX=`TABENV BVPTAD05 PAC7MX $TABTMP/MX`
export PAC7MX
PAC7ET=`TABENV BVPTAD05 PAC7ET $TABUSERS/CDT1ETD05.txt`
export PAC7ET
TABMSG 1009 "BVPTAD05"
rtstab BVPTAD05
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAD05"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
PAC7TS=`TABENV BVPTAD10 PAC7TS \`dirname $TABUSERS\`/TD`
export PAC7TS
PAC7MX=`TABENV BVPTAD10 PAC7MX $TABTMP/MX`
export PAC7MX
PAC7TX=`TABENV BVPTAD10 PAC7TX \`dirname $TABUSERS\`/MBCDT2.txt`
export PAC7TX
PAC7ET=`TABENV BVPTAD10 PAC7ET $TABUSERS/CDT1ETD10.txt`
export PAC7ET
TABMSG 1009 "BVPTAD10"
rtstab BVPTAD10
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAD10"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

CDT2 - Description of steps

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

Code	Physical name	Type	Label
PAC7TD	User dir.: TD	Input	Tables descriptions Slave file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TX	User dir.: MBCDT2.txt	Input	Result extracted from comparison in the CDT1 procedure
PAC7TV	User dir.: TV	Output	File of table-contents associated to the tables descriptions Slave file
PAC7ET	User dir.: CDT2ETD20.txt	Report	Update report

CDT2 - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) CDT2 BATCH PROCEDURE
# * -------------------------------------------------------
# *    VA PACTABLES
# *
# * -------------------------------------------------------
# *    - DISPATCHED TABLE MANAGEMENT -
# *    - UPDATE OF TABLE DESCRIPTIONS AFTER COMPARISON -
# * -------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "-----------------------------------------------------------"
TABMSG 1004 "CDT2"
echo "                          =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "-----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TE.ini
PAC7TD=`TABENV BVPTAD20 PAC7TD \`dirname $TABUSERS\`/TD`
export PAC7TD
PAC7TV=`TABENV BVPTAD20 PAC7TV \`dirname $TABUSERS\`/TV`
export PAC7TV
PAC7TX=`TABENV BVPTAD20 PAC7TX \`dirname $TABUSERS\`/MBCDT2.txt`
export PAC7TX
PAC7ET=`TABENV BVPTAD20 PAC7ET $TABUSERS/CDT2ETD20.txt`
export PAC7ET
TABMSG 1009 "BVPTAD20"
rtstab BVPTAD20
RETURN=$?
```

```
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAD20"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# *************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

CVTA - Table contents update

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

Execution condition

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

CVTA - User input

- One '*'-type line per user:

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

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

Pos.	Len.	Value	Meaning
1	1	'S'	Transaction code
2	1	'A'	Line code
3	6	tttttt	Table number
9	8	DDMMCCYY	Update date: beginning
17	2		Not used
19	1	'/'	Delimiter
20	1		Not used
21	8	DDMMCCYY	Update date: end

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

CVTA - Description of steps

Table-contents comparison: PTAV10

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TV	Base dir.: TV	Input	Tables contents file
PAC7TG	Base dir.: TG	Input	ser parameter file
PAC7MV		Input	Comparison requests
PAC7EX	Tmp dir.: EX	Output	Comparison result
PAC7ET	User dir.: CVTAETV10.txt	Report	Transaction report

Extraction of update transactions: PTAV20

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables Descriptions file
PAC7EX	Tmp dir.: EX	Input	Comparison result
PAC7NU	User dir.: MBUPTA.txt	Output	Update transactions for use as input of UPTA
PAC7ET	User dir.: CVTAETV20.txt	Report	Printing of extracted transactions

CVTA - Execution script

```
#!/bin/sh
#@(#)VA PACTABLES XXX XXX (R) CVTA BATCH PROCEDURE
# * -------------------------------------------------------
# *    VA PACTABLES
# *
# * -------------------------------------------------------
# *           - DIRECT CONSULTATION OF TABLES -
# *
# * -------------------------------------------------------
# *
# * INPUT
# * --> ONE "*" TYPE LINE PER USER
# * COL 2      : "*" : LINE CODE
# * COL 3-10   : USER CODE (UUUUUUUU)
# * COL 11-18  : PASSWORD (PPPPPPPP)
# *
# * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
# * COL 1      : "S" : TRANSACTION CODE
```

```
# * COL 2      : "A" : LINE CODE
# * COL 3-8    : TABLE NUMBER (TTTTTT)
# * COL 9-16   : UPDATE DATE : BEGINNING (DDMMCCYY)
# * COL 17-18  : NOT USED
# * COL 19     : "/" : DELIMITER
# * COL 20     : NOT USED
# * COL 21-28  : UPDATE DATE : END (DDMMCCYY)
# *
# * ------------------------------------------------------
# Parameter control
. $TABDIR/system/proc/TABINIT.ini
echo ""
echo "----------------------------------------------------------"
TABMSG 1004 "CVTA"
echo "                         =============="
TABMSG 1047 "$TABBASE"
TABMSG 1005 "$TABDIR/config/$1"
TABMSG 1006 "$TABTMP"
TABMSG 1073 "$TABUSERS"
TABMSG 1007 "$TABINPUT"
echo "----------------------------------------------------------"
echo ""
TABPAUSE
TABMKDIR
. $TABDIR/config/$1/PAC7TD.ini
. $TABDIR/config/$1/PAC7TE.ini
. $TABDIR/config/$1/PAC7TV.ini
. $TABDIR/config/$1/PAC7TG.ini
PAC7MV=$TABINPUT
export PAC7MV
PAC7EX=`TABENV PAC7EX BVPTAV10 $TABTMP/EX`
export PAC7EX
PAC7ET=`TABENV PAC7ET BVPTAV10 $TABUSERS/CVTAETV10.txt`
export PAC7ET
TABMSG 1009 "BVPTAV10"
rtstab BVPTAV10
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAV10"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# **************************************************************
. $TABDIR/config/$1/PAC7TD.ini
PAC7EX=`TABENV PAC7EX BVPTAV20 $TABTMP/EX`
export PAC7EX
PAC7NU=`TABENV BVPTAD20 PAC7NU \`dirname $TABUSERS\`/MBUPTA.txt`
export PAC7NU
PAC7ET=`TABENV PAC7ET BVPTAV20 $TABUSERS/CVTAETV20.txt`
```

```
export PAC7ET
TABMSG 1009 "BVPTAV20"
rtstab BVPTAV20
RETURN=$?
case $RETURN in
0)
 ;;
*)
 TABMSG 1012 "BVPTAV20"
 TABMSG 1025
 TABERR
 TABRMTMP
 exit $RETURN
 ;;
esac
# ************************************************************
TABMSG 1010
TABRMTMP
exit $RETURN
```

IBM ®

Part Number: DETIX000351A - 7258

Printed in USA