

VisualAge Pacbase



Pactables - Windows 2000 or NT

Version 3.5



VisualAge Pacbase



Pactables - Windows 2000 or NT

Version 3.5

Note

Before using this document, read the general information under “Notices” on page v.

You may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

<http://www.ibm.com/support/docview.wss?rs=37&uid=swg27005477>

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

Second Edition (April 2007)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at: <http://www.ibm.com/software/awdtools/vapacbase/support.html> or to the following postal address:

IBM Paris Laboratory
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.

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

© Copyright International Business Machines Corporation 1983,2007. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v	INTA - Introduction	23
Trademarks	vii	INTA - User input	24
Chapter 1. Foreword	1	INTA - Description of steps	24
Chapter 2. Pactables components	3	INTA - Execution script	24
Introduction	3	GETT - Table generation	25
The macro-structures of TUF-TP facility	3	GETT - Introduction	25
The system files	4	GETT - Description of steps	26
The evolving files	4	GETT - Execution script	26
Standard limitations.	6	UPTA - Table update	27
User access facilities.	6	UPTA - Introduction	27
Chapter 3. Installation	7	UPTA - User input	28
Prerequisites	7	UPTA - Description of steps	29
Hardware and software	7	UPTA - Execution script	30
Disk space	7	PRTA - Table printing	31
Installation	7	PRTA - Introduction	31
Introduction	7	PRTA - User input	32
System installation	8	PRTA - Description of steps	33
Repository installation	9	PRTA - Execution script	34
Pactables database	9	IMTA - Table import	36
Configuration parameters	9	IMTA - Introduction	36
Unreference a database.	9	IMTA - User input	37
Connection	10	IMTA - Description of steps	37
Server start-up and stop	10	IMTA - Execution script	38
3270 emulator start-up	11	RETA - Table reorganization.	40
Use tests	12	RETA - Introduction	40
Chapter 4. Re-installation/removing of components	15	RETA - User input	40
Introduction	15	RETA - Description of steps	41
Procedure start-up	15	RETA - Execution script	43
Chapter 5. The batch procedures	17	SVTA - Backup	45
Introduction	17	SVTA - Introduction	45
Classification of procedures	17	SVTA - Description of steps	45
Abnormal executions	18	SVTA - Execution script	45
List of 'Run Time Errors'.	19	TCTA - Pactables transfer from another platform	46
Management of errors in the procedures	20	TCTA - Introduction	46
Procedure definition	20	TCTA - Description of steps	48
Procedure launch	22	TCTA - Execution script	49
INTA - Table initialization	23	RSTA - Restoration	50
		RSTA - Introduction	50
		RSTA - Description of steps	51
		RSTA - Execution script	51
		LDTA - List of table descriptions	52
		LDTA - Introduction	52
		LDTA - User input	52
		LDTA - Description of steps	53

LDTA - Execution script	53	TUTA - Description of steps.	62
PMTA - Parameter update	54	TUTA - Execution script	62
PMTA - Introduction	54	Dispatched table management (DTM option)	63
PMTA - User input.	54	CDT1-CDT2 - Table descriptions	
PMTA - Description of steps	56	comparison	63
PMTA - Execution script	56	CDT1 - User input	64
EXTA - Table extraction	58	CDT1 - Description of steps.	64
EXTA - Introduction	58	CDT1 - Execution script	65
EXTA - User input	58	CDT2 - Description of steps.	66
EXTA - Description of steps	59	CDT2 - Execution script	66
EXTA - Execution script	60	CVTA - Update of Table contents	67
TUTA - Direct consultation of tables	61	CVTA - User input.	67
TUTA - Introduction	61	CVTA - Description of steps.	68
TUTA - User input.	62	CVTA - Execution script	69

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.

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 installation of the 3.5 version must be distinct from any former Pactables version regarding the installation parameters. The test case provided on the installation media must be executed.

The installation of this version does not require any retrieval.

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 call the BVPFT90 sub-program of 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 into the VA Pac library used for the development of user transactions by taking the transactions of VA Pac UPDP or UPDT procedures as input.

The system files

They represent the system itself. They are not subjected to daily handling and must be re-loaded if the system has to be re-installed. They are constituted of:

- Batch and on-line executable programs
(installed in the SYS/PGM directories)
- a file containing the error messages and the automatic documentation of the Pactables function : TE

Characteristics	Value
Size	About 900 records i.e. about 1,2 Mega-bytes
Organization	Indexed
Length	90
Key	17 (position 1)
Utilization	Batch / on-line
Location	SYS/SKEL directory
Internal name	PAC7TE

The evolving files

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

The first two make up the actual Pactables files:

The Table Description file (TD)

Characteristics	Value
Organization	Indexed
Length	240

Characteristics	Value
Key	21 (position 1)
Utilization	Batch/on-line
Location	DATA/"Database name"/BASE
Internal name	PAC7TD

The Table Contents file (TV)

Characteristics	Value
Organization	Indexed
Length	80 to 1100
Key	35 (position 5)
Utilization	Batch/on-line
Location	DATA/"Database name"/BASE
Internal name	PAC7TV

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

The user parameters file (TG)

Characteristics	Value
Organization	Indexed
Length	85
Key	22 (position 1)
Utilization	Batch/on-line
Location	DATA/"Database name"/BASE
Internal name	PAC7TG

The TG file includes the user codes and the corresponding access authorizations.

The fourth file constitutes the backup of the user Pactables files described above.

The backup file (TC)

Characteristics	Value
Organization	Sequential
Length	1,061
Utilization	Batch
Location	DATA/"Database name"/SAVE
Internal name	PAC7TC

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
BVPFT00	TUF facility
BVPFT90	TUF facility

These programs, supplied with a '.DLL' extension and compiled in the Pactables Windows/NT environment, can be executed with the same Micro Focus Run-Time.

They are supplied in the \$TABDIR/SYS/PGM directory.

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

Chapter 3. Installation

Prerequisites

Hardware and software

- Architecture: a Windows/NT, Windows 2000, Windows XP or Windows 2003 server.
- Processor: workstation with Windows NT server (4.0 minimum), Windows 2000, Windows XP or Windows 2003.
- Memory: 96 Mb. Additional memory may be needed depending on the number of servers installed on the same machine.
- Software:
 - Microsoft Windows Script, version 5.1 onwards. You can download it from the following URL:
www.microsoft.com/downloads
- For the MICROFOCUS version:
MICRO FOCUS Application Server
- For the Acucorp version:
ACUCOBOL-GT

CAUTION:

Besides installing the cobol runtime, you must update the system environment variables (PATH, COBPATH...).

For AcuCobol, the PATH system variable must be completed with the path of the AcuCobol runtime (usually ..\AcuGT\bin). The machine must then be rebooted.

- Installation medium : CD-ROM drive.

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.

Installation

Introduction

The product is supplied on a CD-Rom.

To perform the installation, you must be connected with NT Administrator's authorizations.

By default, the Server is installed on

C:\Program Files\IBM\VisualAge Pacbase\

However you will be able to modify the unit and directory name during the installation process.

The installation is performed in the \Pactables directory and includes two successive steps:

1. installation of the System, which creates the \SYS\ directory
2. installation of the Pactables database, which creates the \CONFIG\[database_name]\ and the \DATA\[database_name]\ subdirectories.

Each step can be executed independently or not, but always in this order.

All the Databases are created with the installation test data and configuration parameters.

CAUTION:

In this manual, [database_name] used in the directory full name corresponds to the name of the directory concerning the database for the server data. It does not correspond to the database code.

System installation

The execution of setup.exe opens a graphical interface which guides you through the installation process.

After a Welcome screen, it displays some workstation characteristics which can be modified:

- Owner's Name and Organization
- Location for the installation (C:\Program Files\IBM\VisualAge Pacbase\ by default)

The installation copies:

- the translation file, the labels file produce by the procedures in case of error or for information,
- the programs,
- the procedures,

- the Database creation utility,
- the startup script models,
- the error messages file.

Repository installation

Pactables database

You can install a Pactables database directly after the system installation, or independently via the 'Start' menu by selecting the [Create New Database] shortcut in the [VisualAge Pacbase 3.5 Pactables Server] group.

To install a Development database, you must specify the following elements:

- the Database name (8 char.),
- the Database code (4 char.),
- the Database port number (between 49152 and 65535, unique for each Database),
- the Database language code (English by default).

More than one Development Database can be installed. each has its own environment.

You can modify the proposed directories (by default: C:\Program Files\IBM\VisualAge Pacbase\).

Therefore, you can modify the backup directory, the temporary files directory, or the user directory which notably contains the procedures execution reports.

Configuration parameters

For each Database, a Configuration directory is created with:

- 1- BvpServer.ini, containing the parameters of the conversational server,

Unreference a database

The BVPDELDB procedure is used to unreference a Database on a Windows platform.

This procedure executes the following operations:

- deletion of the Windows service for a given database,
- deletion of configuration information (CONFIG directory),
- deletion of information specific to the Database in the Windows Registry,

- deletion of shortcuts for the Database,
- renaming of the Database data directories.

Connection

Server start-up and stop

The Listener must be installed in NT Service mode.

Terminals can then connect to Pactables.

Via the 'Start [Database_name] Database Service' shortcut located in the [VisualAge Pacbase 3.5 Pactables Server] programs group of the Start menu, you can start the listener on the [Database_name] Database.

Via the 'Stop [Database_name] Database Service' shortcut located in the [VisualAge Pacbase 3.5 Pactables Server] programs group of the Start menu, you can stop the listener on the [Database_name] Database.

The listener processing parameters are defined in the "Server.wsf" procedure.

When abnormal executions occur in a listener, messages can be displayed in the list of events of the 'Event Viewer' administration tool, under 'Log/Application'.

These 'ERROR'-category events can give a first indication. They generally correspond to problems in the execution environment.

However, in case of abnormal processing, the product support may ask you to activate the 'trace mode' to detect the source of the error.

- TRACE MODE

Different trace levels can be implemented:

- Level 1
Minimum trace allowing to follow the listener processing with the calls to the COBOL communication monitor,
- Level 2
Detailed trace of the listener processing,
- Level 4
Trace of the messages between the listener and the client workstation.

In the listener startup procedure, 'server.wsf', the purpose of the SRV_TRACE environment variable is to activate the trace mode. To use another trace level, you must set the SRV_TRACE variable and re-start the listener.

EXAMPLE:

SRV_TRACE=1 for a trace level 1

SRV_TRACE=3 for a trace level 1 and 2

SRV_TRACE=5 for a trace level 1 and 4

There are two types of trace result files:

- srv[process_number].txt

to trace the listener (BvpServer.exe).

- dial[process_number].txt

to trace each listener connection (BvpDial.exe).

These files are located in the SRV_DIR directory, set by default in server.wsf file to:

.../data/[database_name]/tmp

The SRV_TRACE_DEL environment variable is set to keep all or a part of the traces created by the listener execution in "dialnnn.txt" :

SRV_TRACE_DEL : "ON" (default value)

to keep traces produced by a processing error only.

SRV_TRACE_DEL : "OFF"

to keep all the traces.

3270 emulator start-up

You can access an on-line server, in character mode, via a 3270 emulator.

To access a database in a 3270 mode via an on-line server, you must configure the emulator by indicating:

- the IP address of the machine where the on-line server is implemented,

- the on-line server listening port number, chosen upon installation 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 automatically set in the 'Server.wsf' procedure when the online server is started up.

Transaction code of the user parameters: 'TABPARM'.

Transaction code for the access to the Pactables function: 'PACTABLES'.

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. BVPTA400 and BVPTA410),
 - Reorganization of TD file (Prog. BVPTA420),
 - Constitution of the TC backup file (Prog. BVPTA430),
 - 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. Re-installation/removing of components

Introduction

The system environment of the Pactables server must be re-installed whenever enhancements to the currently installed version are being delivered.

Generally, system files (error messages, generation skeletons, ...) and programs only are impacted by a new subversion.

Procedure start-up

The Pactables Installation CD-Rom also allows you to:

- "Update" an already installed Pactables module.
- "Remove" an already installed Pactables module.

You can also use the Windows service 'Add/Remove' Programs in the Configuration Panel.

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 a return code '12', described by a message on the .Log file of the procedure.

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 Intb.vbs.

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

The summary table below lists the most common values for 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 product 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 Errors'

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-0 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 next 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.

Procedure definition

A procedure is a Windows Scripting script (.wsf), which includes Visual Basic Script scripts (.vbs).

Each procedure can be executed only via a launch Script (.wsf) which contains the following information:

- a <resource id> field constituted of input data (141 lines maximum) or the definition of an input file by WshEnv("BVP_Input") = "full file name" (the priority value),
- the user code,
- the Database external name (directory under \DATA),
- the complete assignments of the configurable files of the procedure, example : WshEnv("PTA310_PAC7NK") = "External table file".

You must enter all this information before launching the procedure.

Other elements can also be specified to assign files or parameterize procedures (e.g. whether the message must be displayed or written in a file).

In any case, the provided user inputs must be verified and adapted to your environment (<resource id> or BVP_Input).

Temporary files, execution reports and output files are found under directories created dynamically.

- Temporary files are located under: \DATA\[Database_name]\TMP\[user_code]\[proc_name]-[number]

They are deleted at the end of the procedure execution. If you do not want these files to be deleted, you must previously set the "DelDo" variable to "NO" in Intb.vbs. So you will have to delete them manually.

Once this variable is set up ("NO" or "YES"), it is valid for ALL the procedures.

- Output files, execution reports are located under: \DATA\[Database_name]\USERS\[user_code]\[proc_name]-[number]

This number is an application execution number ; by default it is the process number of the procedure.

WARNING:

In Pactables for Acucorp, when a procedure containing a Cobol sort is abnormally executing, the temporary sort files may not be automatically deleted. These 16 files, which name begins by "T" are located in the Windows temporary directory (\TMP or \TEMP) or directly under "C:\". Their date and time correspond to the procedure launch time.

Procedure launch

You can execute the launch script (BVPproc.wsf) or the supervisor (Prbvp.vbs):

- via a command line,
- by double-clicking on them,
- via the 'Start' menu: Access the '[Database_name] Database Utilities' shortcut in the 'VisualAge Pacbase 3.5 Pactables Server' group of [Database_name] and enter [Script_name].

NB: The Prbvp.vbs supervisor is an independent VBScript procedure which starts the execution of the launch scripts. It requires 2 arguments: [Database_name] [script_name] ; [Database_name] is located under directory \DATA.

To be executed, each procedure requires the definition of various elements:

1. either in the launch script: This launch script, BVPxxxx.wsf, is in Windows Scripting.
 - a. user code
 - b. Database external name (directory under \DATA)
 - c. The input data is described in the <resource id> field of the script and retrieved in the BVP_Resource environment variable (in this case, the data volume is limited to 141 lines), or can also be contained in a file named in the BVP_Input environment variable (in this case, the data volume is not limited).
 - d. You can assign files, other than the input file, 'in substitution' by defining an environment variable with:
 - . "StepName_FileName" for a substituted assignment in a particular step,
 - . or "ProcName_FileName" for a substituted assignment in the whole procedure.For example, you define the 'WshEnv("PTA310_PAC7NK") = Fil_In with pathname' variable to indicate a substituted assignment in the PTA310 step of the IMTA procedure.
 - e. Information or error messages are:
 - . either displayed on the screen (MsgTyp = 1),
 - . or edited in a file (MsgTyp =2). This is the default value. You parameterize this in the BVP_Msg environment variable.

WARNING: Pactables does not manage errors and missing elements in the launch script.

2. or in the part which is common to all procedures. This common part, named 'INIT.vbs', is a VBScript inclusion, in each procedure.
 - a. Retrieval of the arguments: [Database] and [user]
 - b. Assignment of 2 parameters with 2 possible values:
DelDo = "YES"/"NO" :
the temporary files are deleted (YES) or not (NO)
NoDisp = "YES"/"NO" :
the Cobol "display" are not displayed (YES) or are displayed by cmd (NO)
 - c. Constitution of the MB file (input data) which contains the user code, password and the possible procedure user input.
If the [user] user code is not found in the launch script, it is searched for in the input file.
If the elements required for the execution of the procedure are absent, an error occurs.

Note:

The input file of the retrieval procedures can be defined in the launch script of the procedure via an environment variable:
WshEnv("NStep_NFile") = "old 2.5 full file name", which enables the substitution in the procedure. The 'USER' parameter is required for the retrieval procedures, except if the procedure accepts a user input file different from the file of the 2.5 version.

The TC file (Database backup) is defined by an environment variable: WshEnv("BVP_SaveName"). This variable allows to create the parameter for the definition of the PAC7TC assignment. If this environment variable is not indicated in the script, the default value is [Dir_SAVE & "\TC].

INTA - Table initialization

INTA - Introduction

This procedure initializes the files which contain 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		Assignments of function keys
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 printed page
58	1		Type of resource control *
		blank	Definition of tables resources in security system
		P	Definition of resources in VA Pac
59	1		Lock of user's code *
		blank	Other user's code authorized
		N	Other user's code not authorized

(*) Specific to other platforms than UNIX and WNT.

INTA - Description of steps

Initialization of files: BVPTAINI

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.: INTAED.txt	Report	Initialization review

INTA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----

```

```

REM *                - TABLE INITIALIZATION -
REM *
REM * -----
REM *
REM * INPUT
REM * COL 1-36      : INSTALLATION LABEL
REM * COL 37       : LANGUAGE VERSION PARAMETER
REM *              "E" ENGLISH
REM *              "F" FRENCH
REM * COL 38       : NOT USED
REM * COL 39-50    : FUNCTION KEYS ASSIGNMENTS
REM * COL 56-57   : "NN" NUMBER OF LINES PER PRINTOUT PAGE
REM *
REM * -----
<job id=INTA>

<script language="VBScript">
MyProc = "INTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTAINI"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MD") = Fic_Input
Call BvpEnv("PTAINI","PAC7ED",Rep_USR & "\INTAED.txt")
Call RunCmdLog ("BVPTAINI")
Call Err_Cod(Return , 0 , "PTAINI")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

GETT - Table generation

GETT - Introduction

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

The MD parameter of the JCL must be valorized with the DSN of the PAC7MD file which is in output of the PACT40 step of the GETA or GETD

procedure if the description is for a Pactables V2.0 onwards. Otherwise, the MD parameter must be valorized with the DSN of the PAC7ND file which is in output of the PACT45 step.

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 as input to the GETT procedure.

User input

Result of GETD or GETA extraction.

GETT - Description of steps

Update of table files: BVPTA250

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.: WTK.tmp	Output	Output file
PAC7ET	User dir.: GETTET250.txt	Report	Input/output errors on files

Printing of descriptions: BVPTA290

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

GETT - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE GENERATION -
REM *
REM * -----
REM *
REM * INPUT : RESULT OF GETA OR GETD PROCEDURE

```

```

REM *
REM * -----
<job id=GETT>

<script language="VBScript">
MyProc = "GETT"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTA250"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MD") = Fic_Input
Call BvpEnv("PTA250","PAC7TK",Rep_TMP & "\WTK.tmp")
Call BvpEnv("PTA250","PAC7ET",Rep_USR & "\GETTET250.txt")
Call RunCmdLog ("BVPTA250")
Call Err_Cod(Return , 0 , "PTA250")

Call Msg_Log (Array("1022" , "PTA290"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA290","PAC7TE",Rep_TMP & "\WTK.tmp")
Call BvpEnv("PTA290","PAC7ID",Rep_USR & "\GETTID290.txt")
Call RunCmdLog ("BVPTA290")
Call Err_Cod(Return , 0 , "PTA290")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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, BVPTA302, is available from Pactables 2.0 onwards.

During updates, the BVPTA300 program may call the user validation routines in order to perform additional controls. The default generation option for these routines is 'without century management'.

From Release 2.0 onwards, the user validation routines are generated with the century-management option. The new program, BVPTA302, must therefore be renamed and used instead of the BVPTA300 program.

In all cases, ALL the user validation 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 be updated:

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

Pos.	Len.	Value	Meaning
		'D'	Deletion
2	1	'V'	Line code
3	1		Continuation line
		' '	First data line
		'-'	Item data continuation
4	77		Table data separated by the delimiter indicated on the 'A'-type line

UPTA - Description of steps

Table update: BVPTA300

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.: WDE.tmp	Output	Print requests
PAC7ET	User dir.: UPTAET.txt	Report	Transaction review
PAC7MT	Tmp dir.: WMT.tmp		Formatted transactions

Formatting for printing: BVPTA350

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table descriptions file
PAC7TV	Base dir.: TV	Input	Table contents file
PAC7DE	Tmp dir.: WDE.tmp	Input	Print request
PAC7ET	Tmp dir.: WET.tmp	Output	Print file
PAC7EX	User dir.: UPTAEX.txt	Report	Statistics on printing

Printing of tables: BVPTA360

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Tables descriptions file
PAC7ET	Tmp dir.: WET.tmp	Input	Print file

Code	Physical name	Type	Label
PAC7EY	User dir.: UPTAEY.txt	Report	Printing of tables

UPTA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE UPDATE -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE
REM * COL 11-18  : PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER TABLE TO UPDATE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM * COL 17     : NOT USED
REM * COL 18     : SUB-SYSTEM NUMBER
REM *           : " " : NO SUB-SYSTEM SPECIFIED
REM *           : 1 TO 0 : SUB-SYSTEM NUMBER
REM * COL 19     : DATA DELIMITER
REM *           : " " : CONSIDER AS "/" BY DEFAULT
REM * --> ONE "V" TYPE LINE TO UPDATE TABLE DATA
REM * COL 1      : ACTION CODE
REM *           : "C" : CREATION
REM *           : "M" : MODIFICATION
REM *           : "D" : DELETION
REM * COL 2      : "V" : LINE CODE
REM * COL 3      : CONTINUATION LINE
REM *           : " " : FIRST DATA LINE
REM *           : "-" : ITEM DATA CONTINUATION
REM * COL 4-80   : TABLE DATA SEPARATED BY THE DELIMITER
REM *           : INDICATED ON THE "A" TYPE LINE
REM *
REM * -----
<job id=UPTA>

<script language="VBScript">
MyProc = "UPTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

```



```

Call Msg_Log (Array("1022" , "PTA300"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\\TE"
WshEnv("PAC7TD") = Rep_BASE & "\\TD"
WshEnv("PAC7TG") = Rep_BASE & "\\TG"
WshEnv("PAC7TV") = Rep_BASE & "\\TV"
WshEnv("PAC7MS") = Fic_Input
Call BvpEnv("PTA300","PAC7DE",Rep_TMP & "\\WDE.tmp")
Call BvpEnv("PTA300","PAC7MT",Rep_TMP & "\\WMT.tmp")
Call BvpEnv("PTA300","PAC7ET",Rep_USR & "\\UPTAET.txt")
Call RunCmdLog ("BVPTA300")
Call Err_Cod(Return , 0 , "PTA300")

Call Msg_Log (Array("1022" , "PTA350"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\\TD"
WshEnv("PAC7TV") = Rep_BASE & "\\TV"
Call BvpEnv("PTA350","PAC7DE",Rep_TMP & "\\WDE.tmp")
Call BvpEnv("PTA350","PAC7ET",Rep_TMP & "\\WET.tmp")
Call BvpEnv("PTA350","PAC7EX",Rep_USR & "\\UPTAEX.txt")
Call RunCmdLog ("BVPTA350")
Call Err_Cod(Return , 0 , "PTA350")

Call Msg_Log (Array("1022" , "PTA360"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\\TD"
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\\WET.tmp")
Call BvpEnv("PTA360","PAC7EY",Rep_USR & "\\UPTAEY.txt")
Call RunCmdLog ("BVPTA360")
Call Err_Cod(Return , 0 , "PTA360")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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 the access to on-line use remains open.

Note

Users may also submit the PRTA procedure on-line. on-line mode. (except for the User "*****", you are advised against creating generally any user code including special characters).

To do so, the script required to submit the execution of the procedure must be entered in the LJ screen.

Here is an example of script:

```

TY LINE LABEL
TJ 111111 <job id = BVPPrta>
TJ 111222 <script language="VBScript">
TJ 111333 Dim WshSh
TJ 111444 Set WshSh = WScript.CreateObject("WScript.Shell")
TJ 111450 Set WshEnv = WshSh.Environment("PROCESS")
TJ 111460 Set Args = Wscript.Arguments
TJ 111470 </script>
TJ 111490 <resource id="PRTA"><![CDATA[
TJ 611111 ]]></resource>
TJ 611222 <script language="VBScript">
TJ 611444 Dim FSO
TJ 611450 Set FSO=CreateObject("Scripting.FileSystemObject")
TJ 611460 Rep = "HKLM\SOFTWARE\IBM" & _
TJ 611470 "\BVP VisualAge Pacbase 3.5\Pactables"
TJ 611490 Rep_PROC = WshSh.RegRead (Rep & "\BVP_SYS\PROC\")
TJ 612111 USER=Args(0) 'user code
TJ 612222 BASE=Args(1) 'database code
TJ 612333 'JOB=ARGS(2) 'Job number
TJ 612444 WshEnv("BVP_Resource") = getresource("PRTA")
TJ 621111 PROC=Rep_PROC & "\prta.wsf"
TJ 621333 CMD=chr(34) & PROC & chr(34) & " " & BASE &
TJ 621444 " " & USER
TJ 621470 RetPrta = WshSh.Run(CMD , 1 , TRUE )
TJ 621490 </script>
TJ 621500 </job>

```

PRTA - User input

- One '*'-type line per user:

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

Pos.	Len.	Value	Meaning
11	8	pppppppp	Password

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

Pos.	Len.	Value	Meaning
1	1		Action code
		'E'	Table printing
		'H'	List of historical accounts
		'L'	List of tables
		'P'	Printing of general parameters
		'S'	List of sub-schemas and sub-systems
		'X'	Table contents with historical accounts
2	1	'A'	Line code
3	6	ttttt	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: BVPTA320

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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

Code	Physical name	Type	Label
PAC7DE	Tmp dir.: WDE.tmp	Output	Print requests
PAC7XE	User dir.: PRTAXE320.txt	Report	Transaction review

Preparation for printing: BVPTA350

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table descriptions file
PAC7TV	Base dir.: TV	Input	Table contents file
PAC7DE	Tmp dir.: WDE.tmp	Input	Print requests
PAC7ET	Tmp dir.: WET.tmp	Output	Print file
PAC7EX	User dir.: PRTAEX350.txt	Report	Statistics on printing

Printing of tables: BVPTA360

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

PRTA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE PRINT -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE
REM * COL 11-18  : PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER TABLE TO BE PRINTED
REM * COL 1      : ACTION CODE
REM *           : "E" : TABLE PRINTING
REM *           : "H" : LIST OF HISTORICAL ACCOUNTS
REM *           : "L" : LIST OF THE TABLES
REM *           : "S" : LIST OF SUB-SCHEMAS AND SUB-SYSTEMS
REM *           : "X" : TABLE CONTENTS WITH HISTORICAL
REM *           :      ACCOUNTS
REM * COL 2      : "A" : LINE CODE

```

```

REM * COL 3-8      : TABLE NUMBER (TTTTTT)
REM * COL 9-16    : HISTORICAL ACCOUNT DATE OR DATE OF THE
REM *             : REFERENCE DESCRIPTION
REM * COL 17      : SUB-SCHEMA SELECTION
REM *             : BLANK : NO SUB-SCHEMA SELECTION
REM *             : 1 TO 0 : SELECTED SUB-SCHEMA NUMBER
REM * COL 18      : SUB-SYSTEM SELECTION
REM *             : BLANK : NO SUB-SYSTEM SELECTION
REM *             : 1 TO 0 : SELECTED SUB-SYSTEM NUMBER
REM * COL 19      : PRINT OPTION OF THE KEY'S DATA ELEMENTS
REM *             : BLANK : CONCATENED DATA ELEMENTS
REM *             : "0"  : SEPARATED DATA ELEMENTS
REM *
REM * -----
<job id=PRTA>

```

```

<script language="VBScript">
MyProc = "PRTA"
</script>

```

```

<script language="VBScript" src="Intb.vbs"/>

```

```

<script language="VBScript">

```

```

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1022" , "PTA320"))

```

```

'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7CA") = Fic_Input
Call BvpEnv("PTA320","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA320","PAC7XE",Rep_USR & "\PRTAXE320.txt")
Call RunCmdLog ("BVPTA320")
Call Err_Cod(Return , 0 , "PTA320")

```

```

Call Msg_Log (Array("1022" , "PTA350"))

```

```

'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
Call BvpEnv("PTA350","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EX",Rep_USR & "\PRTAEX350.txt")
Call RunCmdLog ("BVPTA350")
Call Err_Cod(Return , 0 , "PTA350")

```

```

Call Msg_Log (Array("1022" , "PTA360"))

```

```

'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EY",Rep_USR & "\PRTAEY360.txt")

```

```
Call RunCmdLog ("BVPTA360")
Call Err_Cod(Return , 0 , "PTA360")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>
```

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 into 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, before executing this procedure, to save 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, BVPTA312, is shipped with Pactables V2.0 onwards.

During updates, the BVPTA310 program may call user control routines in order to perform additional controls. The default generation option for these routines is 'without century management'.

With the V2.0 onwards, if the user control routines are generated with the century-management option, the new program, BVPTA312, must therefore be renamed and used instead of the BVPTA310 program.

In all cases, ALL the user control 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: BVPTA310

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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	Table contents files
PAC7MV		Input	Request transactions
PAC7NK	User dir.: TABF	Input	External table file
PAC7DE	Tmp dir.: WDE.tmp	Output	Print requests

Code	Physical name	Type	Label
PAC7ET	User dir.: IMTAET310.txt	Report	Execution report

Formatting of printout: BVPTA350

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

Printing: BVPTA360

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

IMTA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE IMPORT -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE
REM * COL 11-18  : PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER TABLE TO BE IMPORTED
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER TO BE IMPORTED (TTTTTT)
REM * COL 9-16   : TABLE DATE (OPTIONAL) (DDMMCCYY)
REM *
REM * -----
<job id=IMTA>

<script language="VBScript">
MyProc = "IMTA"
</script>

```



```

<script language="VBScript" src="Intb.vbs"/>
<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTA310"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTA310","PAC7NK",RepT_USR & "\TABF")
Call BvpEnv("PTA310","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA310","PAC7ET",Rep_USR & "\IMTAET310.txt")
Call RunCmdLog ("BVPTA310")
Call Err_Cod(Return , 0 , "PTA310")

Call Msg_Log (Array("1022" , "PTA350"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
Call BvpEnv("PTA350","PAC7DE",Rep_TMP & "\WDE.tmp")
Call BvpEnv("PTA350","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA350","PAC7EX",Rep_USR & "\IMTAEX350.txt")
Call RunCmdLog ("BVPTA350")
Call Err_Cod(Return , 0 , "PTA350")

Call Msg_Log (Array("1022" , "PTA360"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTA360","PAC7ET",Rep_TMP & "\WET.tmp")
Call BvpEnv("PTA360","PAC7EY",Rep_USR & "\IMTAEY360.txt")
Call RunCmdLog ("BVPTA360")
Call Err_Cod(Return , 0 , "PTA360")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

RETA - Table reorganization

RETA - Introduction

From the Pactables Database, this procedure rebuilds the backup file which contains the new tables' description and content files; these files are the 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 via an 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	tttttt	Table number
9	8	DDMMCCYY	Historical account date
19	1	' '	Option
			- when the action code is equal to 'G', the historical account whose date is equal to the date specified is kept. If there is no date, all historical accounts are kept.

Pos.	Len.	Value	Meaning
			- 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: BVPTA400

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table descriptions file
PAC7TE	System - Skel dir.: TE	Input	Error-message file
PAC7TG	Base dir.: TG	Input	User parameter file
			Table contents file
	Base dir.: TV		
PAC7DR		Input	Reorganization requests
PAC7TX	Tmp dir.: WTX.tmp	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 to 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: BVPTA410

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

Reorganization of table descriptions: BVPTA420

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

Note

The PAC7ML file (Table description print request) must be kept and used as input to the LDTA procedure, to output a printing of the table descriptions that were kept, in order to check the correct execution of the reorganization.

Building of backup file: BVPTA430

Code	Physical name	Type	Label
PAC7TW	Tmp dir.: WTW.tmp	Input	Validated contents file
PAC7TS	Tmp dir.: WTS.tmp	Input	Reorganized descriptions file
PAC7TC	Save dir.: TC-new	Output	Backup file resulting from reorganization

TG file backup: BVPTASVG

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

RETA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - TABLE REORGANIZATION -
REM *
REM * -----
REM * INPUT
REM * --> ONE "*" TYPE LINE IDENTIFYING PACTABLES MANAGER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : PACTABLES MANAGER CODE
REM * COL 11-18  : PACTABLES MANAGER PASSWORD
REM *
REM * --> ONE "A" TYPE LINE PER HISTORICAL ACCOUNT TO KEEP
REM *   OR DELETE
REM * COL 1      : ACTION CODE
REM *           : "S" : HISTORICAL ACCOUNT TO PURGE
REM *           : "G" : HISTORICAL ACCOUNT TO KEEP
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM * COL 19     : OPTION
REM *           : " " : SEE DOCUMENTATION FOR MORE
REM *           : "X" : EXPLANATIONS
REM *           : "X"
REM *
REM * -----
<job id=RETA>

<script language="VBScript">
MyProc = "RETA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTA400"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\\TE"
WshEnv("PAC7TD") = Rep_BASE & "\\TD"
WshEnv("PAC7TG") = Rep_BASE & "\\TG"

```

```

WshEnv("PAC7TV") = Rep_BASE & "\\TV"
WshEnv("PAC7DR") = Fic_Input
Call BvpEnv("PTA400","PAC7TX",Rep_TMP & "\\WTX.tmp")
Call BvpEnv("PTA400","PAC7DE",Rep_USR & "\\MVRETA.txt")
Call BvpEnv("PTA400","PAC7IR",Rep_USR & "\\RETAIR400.txt")
Call RunCmdLog ("BVPTA400")
Call Err_Cod(Return , 0 , "PTA400")

Call Msg_Log (Array("1022" , "PTA410"))
'-----
Call BvpEnv("PTA410","PAC7TX",Rep_TMP & "\\WTX.tmp")
Call BvpEnv("PTA410","PAC7TW",Rep_TMP & "\\WTW.tmp")
WshEnv("PAC7MB") = Fic_Input
Call RunCmdLog ("BVPTA410")
Call Err_Cod(Return , 0 , "PTA410")

Call Msg_Log (Array("1022" , "PTA420"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\\TD"
Call BvpEnv("PTA420","PAC7TS",Rep_TMP & "\\WTS.tmp")
Call BvpEnv("PTA420","PAC7DE",Rep_USR & "\\MVRETA.txt")
Call BvpEnv("PTA420","PAC7ML",Rep_USR & "\\MWRETA.txt")
Call RunCmdLog ("BVPTA420")
Call Err_Cod(Return , 0 , "PTA420")

Call Msg_Log (Array("1022" , "PTA430"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\\TC-new"
Call BvpEnv("PTA430","PAC7TW",Rep_TMP & "\\WTW.tmp")
Call BvpEnv("PTA430","PAC7TS",Rep_TMP & "\\WTS.tmp")
Call RunCmdLog ("BVPTA430")
Call Err_Cod(Return , 0 , "PTA430")

Call Msg_Log (Array("1022" , "PTASVG"))
'-----
WshEnv("PAC7TG") = Rep_BASE & "\\TG"
WshEnv("PAC7TC") = Rep_SAVE & "\\TC-new"
Call RunCmdLog ("BVPTASVG")
Call Err_Cod(Return , 0 , "PTASVG")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\\TC")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

SVTA - Backup

SVTA - Introduction

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

Execution condition

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

User input

None.

SVTA - Description of steps

TD backup: BVPTASVT

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

TV backup: BVPTASVV

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

TG backup: BVPTASVG

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

SVTA - Execution script

```
REM * -----  
REM *   VA PACTABLES  
REM *  
REM * -----  
REM *           - TABLES BACKUP -  
REM *  
REM * -----
```

```

<job id=SVTA>

<script language="VBScript">
MyProc = "SVTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTASVT"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call RunCmdLog ("BVPTASVT")
Call Err_Cod(Return , 0 , "PTASVT")

Call Msg_Log (Array("1022" , "PTASVV"))
'-----
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call RunCmdLog ("BVPTASVV")
Call Err_Cod(Return , 0 , "PTASVV")

Call Msg_Log (Array("1022" , "PTASVG"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
Call RunCmdLog ("BVPTASVG")
Call Err_Cod(Return , 0 , "PTASVG")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\TC")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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 version of the source site is the same as that of the target site, the actions to be performed are the following ones:

- Backup on the source site (SVTA procedure)
- Transfer of the TC file produced by SVTA to the target platform,
- Retrieval of the file on the target platform (TCTA procedure),
- Restoration of the database (RSTA procedure), using as 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: BVPTATC1

Code	Physical name	Type	Label
PAC7TC	Save dir.: TC.INI	Input	Input backup file
PAC7SD	Tmp dir.: WSD.tmp	Output	Sequential image of table descriptions
PAC7SV	Tmp dir.: WSV.tmp	Output	Sequential image of table contents
PAC7SG	Tmp dir.: WSG.tmp	Output	Parameter sequential image

Sort of Table descriptions: BVPTATCD

Code	Physical name	Type	Label
PAC7SD	Tmp dir.: WDS.tmp	Input	Sequential image of table descriptions
PAC7AD	Tmp dir.: WAD.tmp	Output	Sorted table descriptions

Sort of Table contents: BVPTATCV

Code	Physical name	Type	Label
PAC7SV	Tmp dir.: WSV.tmp	Input	Sequential image of table contents
PAC7AV	Tmp dir.: WAV.tmp	Output	Sorted table contents

Sort of User parameters: BVPTATCG

Code	Physical name	Type	Label
PAC7SG	Tmp dir.: WSG.tmp	Input	Sequential image of parameters
PAC7AG	Tmp dir.: WAG.tmp	Output	Sorted user parameters

Reconstitution of the TC backup: BVPTATC2

Code	Physical name	Type	Label
PAC7AD	Tmp dir.: WAD.tmp	Input	Sequential image of table descriptions
PAC7AV	Tmp dir.: WAV.tmp	Input	Sequential image of contents
PAC7AG	Tmp dir.: WAG.tmp	Input	Sequential image of parameters
PAC7TC	Save dir.: TC-new	Output	TC backup in EBCDIC format

TCTA - Execution script

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *   - TABLES TRANSFER FROM ANOTHER PLATFORM -
REM *
REM * -----
<job id=TCTA>

<script language="VBScript">
MyProc = "TCTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTATC1"))
'-----
WSHENV("PAC7TC") = REP_SAVE & "\TC.INI"
Call BvpEnv("PTATC1","PAC7SD",Rep_TMP & "\WSD.tmp")
Call BvpEnv("PTATC1","PAC7SG",Rep_TMP & "\WSG.tmp")
Call BvpEnv("PTATC1","PAC7SV",Rep_TMP & "\WSV.tmp")
Call RunCmdLog ("BVPTATC1")
Call Err_Cod(Return , 0 , "PTATC1")

Call Msg_Log (Array("1022" , "PTATCD"))
'-----
Call BvpEnv("PTATC1","PAC7SD",Rep_TMP & "\WSD.tmp")
Call BvpEnv("PTATC1","PAC7AD",Rep_TMP & "\WAD.tmp")
Call RunCmdLog ("BVPTATCD")
Call Err_Cod(Return , 0 , "PTATCD")

Call Msg_Log (Array("1022" , "PTATCV"))
'-----
Call BvpEnv("PTATC1","PAC7SV",Rep_TMP & "\WSV.tmp")
Call BvpEnv("PTATC1","PAC7AV",Rep_TMP & "\WAV.tmp")
Call RunCmdLog ("BVPTATCV")
Call Err_Cod(Return , 0 , "PTATCV")

Call Msg_Log (Array("1022" , "PTATCG"))
'-----
Call BvpEnv("PTATC1","PAC7SG",Rep_TMP & "\WSG.tmp")
Call BvpEnv("PTATC1","PAC7AG",Rep_TMP & "\WAG.tmp")
Call RunCmdLog ("BVPTATCG")
Call Err_Cod(Return , 0 , "PTATCG")

Call Msg_Log (Array("1022" , "PTATC2"))
'-----
```

```

WshEnv("PAC7TC") = Rep_SAVE & "\TC-new"
Call BvpEnv("PTATC2","PAC7AD",Rep_TMP & "\WAD.tmp")
Call BvpEnv("PTATC2","PAC7AG",Rep_TMP & "\WAG.tmp")
Call BvpEnv("PTATC2","PAC7AV",Rep_TMP & "\WAV.tmp")
Call RunCmdLog ("BVPTATC2")
Call Err_Cod(Return , 0 , "PTATC2")

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\TC")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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: BVPTARST

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

Restoration of TV: PTARSV

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

Restoration of TG: BVPTARSG

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

RSTA - Execution script

```
REM * -----  
REM *   VA PACTABLES  
REM *  
REM * -----  
REM *           - TABLES RESTORATION -  
REM *  
REM * -----  
<job id=RSTA>  
  
<script language="VBScript">  
MyProc = "RSTA"  
</script>  
  
<script language="VBScript" src="Intb.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTARST"))  
'-----  
WshEnv("PAC7TD") = Rep_BASE & "\TD"  
WshEnv("PAC7TC") = Rep_SAVE & "\TC"  
Call RunCmdLog ("BVPTARST")  
Call Err_Cod(Return , 0 , "PTARST")
```

```

Call Msg_Log (Array("1022" , "PTARSV"))
'-----
WshEnv("PAC7TV") = Rep_BASE & "\\TV"
WshEnv("PAC7TC") = Rep_SAVE & "\\TC"
Call RunCmdLog ("BVPTARSV")
Call Err_Cod(Return , 0 , "PTARSV")

Call Msg_Log (Array("1022" , "PTARSG"))
'-----
WshEnv("PAC7TC") = Rep_SAVE & "\\TC"
WshEnv("PAC7TG") = Rep_BASE & "\\TG"
Call RunCmdLog ("BVPTARSG")
Call Err_Cod(Return , 0 , "PTARSG")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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: BVPTA290

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

LDTA - Execution script

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - LIST OF TABLE DESCRIPTIONS -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "Z" TYPE LINE PER PRINT REQUEST
REM * COL 2      : "Z" : LINE CODE
REM * COL 5-8    : PRINT REQUEST
REM *           : "TLS" : LIST OF TABLE DESCRIPTIONS
REM *           : "TDS" : TABE DESCRIPTION
REM * COL 9-14   : TABLE NUMBER (TTTTTT)
REM * COL 23-30  : HISTORICAL ACCOUNT DATE (MMDDCCYY)
REM *
REM * -----
<job id=LDTA>

<script language="VBScript">
MyProc = "LDTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTA290"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TE") = Fic_Input
Call BvpEnv("PTA290","PAC7ID",Rep_USR & "\LDTAID290.txt")
Call RunCmdLog ("BVPTA290")
Call Err_Cod(Return , 0 , "PTA290")

Call Msg_Log (Array("1023"))
'-----
```

```
Call DeleteFldr(Rep_TMP)
```

```
</script>
```

```
</job>
```

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 hardware 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, the 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

Update of user parameters: BVPTA100

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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.: WNU.tmp	Output	Parameter printing requests
PAC7ET	User dir.: PMTAET100.txt	Report	Printing of descriptions

Printing of user parameters: BVPTA120

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

PMTA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - USER PARAMETER UPDATE -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "TA" LINE : USER PARAMETER UPDATING
REM * COL 1           : ACTION CODE
REM *                 : " " : CREATION OR MODIFICATION
REM *                 : "C" : CREATION
REM *                 : "M" : MODIFICATION
REM *                 : "D" : DELETION
REM * COL 2-9         : USER CODE (UUUUUUUU)
REM * COL 10-11      : "TA" : LINE CODE
REM * COL 12-19     : PASSWORD (PPPPPPPP)

```

```

REM * COL 20      : GENERAL ACCESS AUHTORIZATION
REM *            : "0" : NO GENERAL ACCESS AUTHORIZATION
REM *            : "1" : READ ONLY ACCESS AUTHORIZATION
REM *            : "2" : READ WRITE AUTHORIZATION ON TABLES
REM *            : "3" : READ WRITE AUTHORIZATION ON USER
REM *            :      CODE
REM *
REM * --> ONE "TC" LINE : ACCESS AUTHORIZATIONS PER TABLE
REM * COL 1        : ACTION CODE
REM *            : " " : CREATION OR MODIFICATION
REM *            : "C" : CREATION
REM *            : "M" : MODIFICATION
REM *            : "D" : DELETION
REM * COL 2-9      : USER CODE (UUUUUUUU)
REM * COL 10-11    : "TC" : LINE CODE
REM * COL 12-17    : TABLE CODE (TTTTTT)
REM * COL 18-20    : LINE NUMBER (NNN)
REM * COL 21-80    : ACCESS AUTHORIZATIONS : 20 ACCESS
REM *            : AUTHORIZATIONS MAY BE ENTERED IN THIS
REM *            : FIELD, WITH, FOR EACH AUTHORIZATION, THE
REM *            : SUB-SCHEMA NUMBER (N), THE SUB-SYSTEM
REM *            : NUMBER (N), THE AUTHORIZATION (0,1,2).
REM *            : "*" IS REQUIRED FOR ALL SUB-SCHEMAS AND
REM *            : SUB-SYSTEMS.
REM *
REM * --> ONE "TJ" LINE : CONTROL CARDS
REM * COL 1        : ACTION CODE
REM *            : " " : CREATION OR MODIFICATION
REM *            : "C" : CREATION
REM *            : "M" : MODIFICATION
REM *            : "D" : DELETION
REM * COL 2-9      : USER CODE (UUUUUUUU)
REM * COL 10-11    : "TJ" : LINE CODE
REM * COL 12-17    : TABLE CODE (TTTTTT)
REM * COL 18-20    : SCRIPT LINE NUMBER
REM *            :LT600000 : CONTROL CARD IN FRONT OF PROGRAM
REM *            :GT599999 : CONTROL CARD IN BACK OF PROGRAM
REM * COL 21-80    : CONTENT OF THE SCRIPT LINE
REM *
REM * -----
<job id=PMTA>

<script language="VBScript">
MyProc = "PMTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTA100"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"

```

```

WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTA100","PAC7NU",Rep_TMP & "\WNU.tmp")
Call BvpEnv("PTA100","PAC7ET",Rep_USR & "\PMTAET100.txt")
Call RunCmdLog ("BVPTA100")
Call Err_Cod(Return , 0 , "PTA100")

Call Msg_Log (Array("1022" , "PTA120"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
Call BvpEnv("PTA120","PAC7NU",Rep_TMP & "\WNU.tmp")
Call BvpEnv("PTA120","PAC7ET",Rep_USR & "\PMTAET120.txt")
Call RunCmdLog ("BVPTA120")
Call Err_Cod(Return , 0 , "PTA120")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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

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

EXTA - Description of steps

Extraction of table data: BVPTA150

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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
PAC7MV		Input	Extraction requests
PAC7EX	Tmp dir.: WEX.tmp	Output	Extracted transactions
PAC7ET	User dir.: EXTAET150.txt	Report	Transaction review

Printing of extracted transactions: BVPTA160

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table descriptions file
PAC7EX	Tmp dir.: WEX.tmp	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

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

<script language="VBScript">
MyProc = "EXTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTA150"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MV") = Fic Input
Call BvpEnv("PTA150","PAC7EX",Rep_TMP & "\WEX.tmp")
Call BvpEnv("PTA150","PAC7ET",Rep_USR & "\EXTAET150.txt")
Call RunCmdLog ("BVPTA150")
Call Err_Cod(Return , 0 , "PTA150")

Call Msg_Log (Array("1022" , "PTA160"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
```

```

Call BvpEnv("PTA160","PAC7EX",Rep_TMP & "\WEX.tmp")

'One for each procedure : Rep_USR
'One for all the procedure : RepT_USR

'Call BvpEnv("PTA160","PAC7NU",Rep_USR & "\NUTAB.txt")
Call BvpEnv("PTA160","PAC7NU",RepT_USR & "\NUTAB.txt")

Call BvpEnv("PTA160","PAC7ET",Rep_USR & "\EXTAET160.txt")
Call RunCmdLog ("BVPTA160")

If Return = 0 Then
Call Msg_Log (Array("1058"))
End If
If Return = 8 Then
Call Msg_Log (Array("1059"))
End If
If Return = 12 Then
Call Msg_Log (Array("1060"))
End If
Call Err_Cod(Return , 0 , "PTA160")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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	tttttt	Table number
9	8	DDMMCCYY	Historical account date

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

TUTA - Description of steps

Direct consultation of tables: BVPTAU80

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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	Table descriptions file
PAC7AV	Base dir.: AV	Output	Tables contents file
PAC7ET	User dir.: TUTAET.txt	Report	Transaction report

TUTA - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - DIRECT CONSULTATION OF TABLES -
REM *
REM * -----
REM *
REM * INPUT

```



```

REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE (UUUUUUUU)
REM * COL 11-18  : PASSWORD (PPPPPPPP)
REM *
REM * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : HISTORICAL ACCOUNT DATE (DDMMCCYY)
REM *
REM * -----
<job id=TUTA>

<script language="VBScript">
MyProc = "TUTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTAU80"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MX") = Fic_Input
WshEnv("PAC7AD") = Rep_BASE & "\AD"
WshEnv("PAC7AV") = Rep_BASE & "\AV"
Call BvpEnv("PTAU80","PAC7ET",Rep_USR & "\TUTAET.txt")
Call RunCmdLog ("BVPTAU80")
Call Err_Cod(Return , 0 , "PTAU80")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

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).

Execution condition

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

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

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: BVPTAD05

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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.: WMX.tmp	Output	Validated comparison request transactions

Code	Physical name	Type	Label
PAC7ET	User dir.: CDT1ETD05.txt	Report	Transaction report

Table-description comparison and extraction: BVPTAD10

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.: WMX.tmp	Input	Validated transactions
PAC7TX	User dir.: MBCDT2.txt	Output	Comparison result to be used as input to the CDT2 procedure
PAC7ET	User dir.: CDT1ETD10.txt	Report	Extraction printout

CDT1 - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *       - DISPATCHED TABLE MANAGEMENT -
REM *       - TABLE DESCRIPTION COMPARISON -
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE (UUUUUUUU)
REM * COL 11-18  : PASSWORD (PPPPPPPP)
REM *
REM * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM *
REM * -----
<job id=CDT1>

<script language="VBScript">
MyProc = "CDT1"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1022" , "PTAD05"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
Call BvpEnv("PTAD05","PAC7TD",Rep_BASE & "\TD")
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTAD05","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PTAD05","PAC7ET",Rep_USR & "\CDT1ETD05.txt")
Call RunCmdLog ("BVPTAD05")
Call Err_Cod(Return , 0 , "PTAD05")

Call Msg_Log (Array("1022" , "PTAD10"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
Call BvpEnv("PTAD10","PAC7TD",Rep_BASE & "\TD")
Call BvpEnv("PTAD10","PAC7TS",Rep_USR & "\TD")
Call BvpEnv("PTAD10","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PTAD10","PAC7TX",Rep_USR & "\MBCDT2.txt")
Call BvpEnv("PTAD10","PAC7ET",Rep_USR & "\CDT1ETD10.txt")
Call RunCmdLog ("BVPTAD10")
Call Err_Cod(Return , 0 , "PTAD10")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

CDT2 - Description of steps

Update of 'slave' files, Recognition of the file extracted by CDT1: BVPTAD20

Code	Physical name	Type	Label
PAC7TD	User dir.: TD	Input	Table descriptions Slave file
PAC7TE	System - Skel dir.: TE	Input	Error message file
PAC7TX	User dir.: MBCDT2.txt	Input	Result extracted from the comparison in the CDT1 procedure
PAC7TV	User dir.: TV	Output	Table contents file associated with the table descriptions Slave file
PAC7ET	User dir.: CDT2ET.txt	Report	Update report

CDT2 - Execution script

```

REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *   - DISPATCHED TABLE MANAGEMENT -
REM *   - UPDATE OF TABLE DESCRIPTIONS AFTER COMPARISON -

```

```

REM * -----
<job id=CDT2>

<script language="VBScript">
MyProc = "CDT2"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTAD20"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
Call BvpEnv("PTAD20", "PAC7TD", Rep_BASE & "\TD")
Call BvpEnv("PTAD20", "PAC7TV", RepT_USR & "\TV")
Call BvpEnv("PTAD20", "PAC7TX", RepT_USR & "\MBCDT2.txt")
Call BvpEnv("PTAD20", "PAC7ET", Rep_USR & "\CDT2ET.txt")
Call RunCmdLog ("BVPTAD20")
Call Err_Cod(Return , 0 , "PTAD20")

Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)

</script>
</job>

```

CVTA - Update of Table contents

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

Execution condition

This procedure reads the Pactables files ; it can be executed even if the files 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: BVPTAV10

Code	Physical name	Type	Label
PAC7TD	Base dir.: TD	Input	Table 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
PAC7MV		Input	Comparison requests
PAC7EX	Tmp dir.: WEX.tmp	Output	Comparison result
PAC7ET	User dir.: CVTAETV10.txt	Report	Transaction report

Extraction of update transactions: BVPTAV20

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

CVTA - Execution script

```
REM * -----
REM *   VA PACTABLES
REM *
REM * -----
REM *           - DIRECT CONSULTATION OF TABLES -
REM *
REM * -----
REM *
REM * INPUT
REM * --> ONE "*" TYPE LINE PER USER
REM * COL 2      : "*" : LINE CODE
REM * COL 3-10   : USER CODE (UUUUUUUU)
REM * COL 11-18  : PASSWORD (PPPPPPPP)
REM *
REM * --> ONE "A" LINE TYPE FOR EACH SELECTED TABLE
REM * COL 1      : "S" : TRANSACTION CODE
REM * COL 2      : "A" : LINE CODE
REM * COL 3-8    : TABLE NUMBER (TTTTTT)
REM * COL 9-16   : UPDATE DATE : BEGINNING (DDMMCCYY)
REM * COL 17-18  : NOT USED
REM * COL 19     : "/" : DELIMITER
REM * COL 20     : NOT USED
REM * COL 21-28  : UPDATE DATE : END (DDMMCCYY)
REM *
REM * -----
<job id=CVTA>

<script language="VBScript">
MyProc = "CVTA"
</script>

<script language="VBScript" src="Intb.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTAV10"))
'-----
WshEnv("PAC7TE") = Rep_SKEL & "\TE"
WshEnv("PAC7TD") = Rep_BASE & "\TD"
WshEnv("PAC7TG") = Rep_BASE & "\TG"
WshEnv("PAC7TV") = Rep_BASE & "\TV"
WshEnv("PAC7MV") = Fic_Input
Call BvpEnv("PTAV10","PAC7EX",Rep_TMP & "\WEX.tmp")
Call BvpEnv("PTAV10","PAC7ET",Rep_USR & "\CVTAETV10.txt")
Call RunCmdLog ("BVPTAV10")
Call Err_Cod(Return , 0 , "PTAV10")

Call Msg_Log (Array("1022" , "PTAV20"))
'-----
WshEnv("PAC7TD") = Rep_BASE & "\TD"
Call BvpEnv("PTAV10","PAC7EX",Rep_TMP & "\WEX.tmp")
```

```
Call BvpEnv("PTAV20","PAC7NU",RepT_USR & "\MBUPTA.txt")
Call BvpEnv("PTAV20","PAC7ET",Rep_USR & "\CVTAETV20.txt")
Call RunCmdLog ("BVPTAV20")
Call Err_Cod(Return , 0 , "PTAV20")
```

```
Call Msg_Log (Array("1023"))
'-----
Call DeleteFldr(Rep_TMP)
```

```
</script>
</job>
```




Part Number: DETNT000352A - 7618

Printed in USA