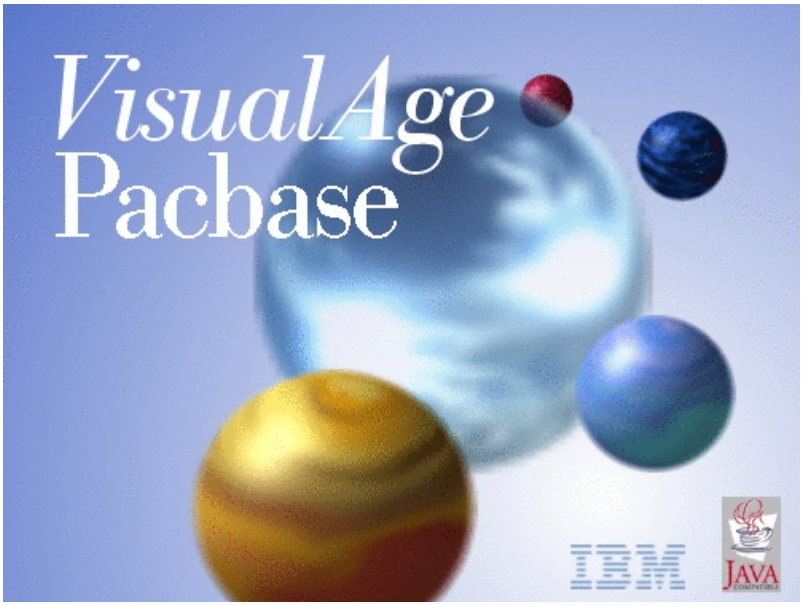
VisualAge Pacbase

DSMS: Installation & Operations Z/OS CICS Server

Version 3.5



VisualAge Pacbase

DSMS: Installation & Operations Z/OS CICS Server

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.

First Edition (July 2006)

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.

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.

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

Use of the manual

This manual is intended for the person in charge of the installation and for the DSMS Database Manager.

It describes the DSMS components, the environment, the batch procedures, the instructions for installing the new version and the operations to be carried out for a standard reinstallation of corrected versions.

Note

DSMS 3.5 requires a complete installation of the technical package, i.e. files, programs and batch procedures.

Chapter 2. DSMS Components

Introduction

DSMS manages permanent data in batch and on-line mode.

Three types of resources are required for the operation of DSMS.

- Libraries which store the DSMS operating programs and system parameters:
  - An on-line program library
  - A batch program library
- Permanent files containing data manipulated by the DSMS system programs previously defined:
  - A system file containing error messages and HELP documentation on DSMS,
  - User files containing the User and Administrator data.
- A library containing the operations parameters.

Note: This manual describes the installation and operation of DSMS. DSMS can be installed independently of other VisualAge Pacbase functions and facilities.

For further details on the operation of the Function itself, refer to the DSMS Reference Manual.

On-line programs

Program	Comment
BVPCHOI	Choice-decoding sub-program
BVPTPDF	DAF extraction sub-program
BVP00AA	Initial screen
BVP00AB	Abend map
BVP00BA	HC
BVP00B1	C .....
BVP00B2	C C
BVP00B3	C ..... Q
BVP00B4	C ..... M

Program	Comment
BVP00B5	XS
BVP00B6	C.....U
BVP00EA	HE
BVP00E1	E .....
BVP00E2	C ..... D E ..... DN/DT
BVP00E3	C ..... F E ..... FN/FT
BVP00E4	C ..... T E ..... T
BVP00E5	LCE
BVP00E6	C ..... S E ..... S
BVP00FA	HPF
BVP00FB	HSC
BVP00HE	Help function
BVP00JO	JO
BVP00KA	HK
BVP00K1	LGK_ LAK
BVP00K2	LPK
BVP00K3	WS WU
BVP00LE	LDE_ LNC_ LSE_ LDC LNC
BVP00LS	LIE .....*...
BVP00MA	H (Main Menu)
BVP00PA	HP
BVP00P1	PL
BVP00QA	HQ
BVP00QB	Q C ..... DD
BVP00QC	R . ..... CD
BVP00Q1	Q . .....
BVP00Q2	Q . ..... D
BVP00Q3	LCQ
BVP00Q4	LVQ
BVP00Q5	LJQ
BVP00Q6	R . .....
BVP00Q7	R L
BVP00Q8	R C

Program	Comment
BVP00Q9	LCR
BVP00SA	HS
BVP00SI	S .....*... U
BVP00S1	S .....*...
BVP00S3	S .....*... V
BVP00S4	S .....*... C
BVP00S5	S .....*... LC
BVP00S6	S .....*... G
BVP00S7	LSS
BVP00S8	LNS LCS
BVP00S9	S .....*... LV
BVP00TA	HT
BVP00TT	TUP
BVP00TU	TRA
BVP00TV	TLA
BVP00TW	TPH
BVP00TX	TUG
BVP00TY	TUS
BVP00TZ	TOP
BVP00T1	TST
BVP00T2	TSU
BVP00T3	TGR
BVP00T4	TPR
BVP00T5	TRE
BVP00T6	TTY
BVP00T7	TUD
BVP00T8	TVE
BVP00T9	TAT
BVPCUAM	Sub-pgm for additional check on BVP00B1
BVPCUEV	Sub-pgm for additional check on BVP00E1
BVPCUMQ	Sub-pgm for additional check on BVP00Q6
BVPCURQ	Sub-pgm for additional check on BVP00Q1
BVPCUSI	Sub-pgm for additional check on BVP00SI

Program	Comment
BVPMONI	Function's 'monitor' program
BVPMOSO	Exit from DSMS transaction
BVPUCTR	Management of the terminal's UCTRAN option
BVPUCTX	Inhibition of the UCTRAN-option management
BVPSECT	Security Systems Interface

Note:

The size of the library is roughly 600 blocks, plus 30 'directory' blocks.

Batch programs

Code	Proc	Comments
BVDAFD10	DPDF	DAF pre-processor
BVPACABE		Batch abend
BVPDCHOI	DUPT	Choice decoding sub-program
BVPDSA10	DPRT	DPRT print sub-program
BVPDSBE	-	DPRT flow monitor (English version)
BVPDSBF	-	DPRT flow monitor (French version)
BVPDSBAS	DSAV	Checks data integrity
BVPDSCAM	DUPT	Additional check sub-program for PDSUB1
BVPDSCEV	-	Additional check sub-program for PDSUE1
BVPDSCMQ	-	Additional check sub-program for PDSUQ6
BVPDSCRQ	-	Additional check sub-program for PDSUQ1
BVPDSCSI	-	Additional check sub-program for PDSUS1
BVPDTPDF	-	DAF extraction sub-program
BVPDSEXE	DEXT	DEXT flow monitor (English version)
BVPDSEXF	DEXT	DEXT flow monitor (French version)
BVPDSE90	DPRT	DPRT print sub-program
BVPDSFAC	-	File-access sub-program
BVPDSINI	DINI	Initializes DSMS files
BVPDSJMS	DREN	Changes the codes in the Journal
BVPDSLVB	DLVB	Replaces low-values with blanks in the BB backup file

Code	Proc	Comments
BVPDSMSE	DREN	Replacement monitor for table codes, keywords and site codes (English version)
BVPDSMSF	-	Same as BVPDSMSE (French version)
BVPDSRCT	DREN	Input transaction check
BVPDSRFU	-	Sorts merges
BVPDSRMS	-	Changes the codes in the backup
BVPDSRQ0	DPRT	Analyzes queries
BVPDSRQ1	-	Selects and extracts queries
BVPDSRQ2	-	Formats elements
BVPDSRQ3	-	Extracts and prints the data
BVPDSR10	DREO	Reorganizes the cross-reference file
BVPDSR20	-	
BVPDSR30	-	
BVPDSR40	DREO	Reorganization
BVPDSUAA	DUPT	Sub-program
BVPDSUB1	-	-
BVPDSUB2	-	-
BVPDSUB3	-	-
BVPDSUB4	DUPT	-
BVPDSUB6	-	-
BVPDSUE1	-	-
BVPDSUE2	-	-
BVPDSUE3	-	Sub-program
BVPDSUK1	-	-
BVPDSUP0	-	Batch update monitor
BVPDSUP1	-	Sub-program
BVPDSUQ1	-	-
BVPDSUQ2	-	-
BVPDSUQ5	-	-
BVPDSUQ6	-	-
BVPDSUQ7	-	-
BVPDSUQ8	-	-
BVPDSUSI	-	-

Code	Proc	Comments
BVPDSUS1	-	-
BVPDSUS3	-	-
BVPDSUS4	-	-
BVPDSUS6	-	-
BVPDSUTT	-	-
BVPDSUTV	-	-
BVPDSUTW	-	-
BVPDSUTX	-	-
BVPDSUTY	-	-
BVPDSUTZ	-	-
BVPDSUT1	-	-
BVPDSUT2	-	-
BVPDSUT3	-	-
BVPDSUT4	-	-
BVPDSUT5	-	-
BVPDSUT6	-	-
BVPDSUT7	-	-
BVPDSUT8	-	-
BVPDSUT9	-	-
BVPDSXCT	DEXT	Checks validity of input
BVPDSXST	DEXT	Sorts entities
BVPDSXTR	-	Extracts entities
BVPDS300	DARC	Archives and deactivates the journal
BVPDS320	-	Reinitializes the journal
BVPDS38E	-	Checks the journal (English version)
BVPDS38F	DRST	Checks the journal (French version)
BVPDS400	-	Restores or initializes the files
BVPDS450	-	Retrieves the archived journal
BVPDS500	DSAV	Saves data/elements/references
BVPDS600	DEXP	Extracts VA Pac journal
BVPDS610	-	
BVPDS700	DXBJ	Extracts journalized transactions
BVPDS900	DUPD	Update of DSMS from DAF tables

Code	Proc	Comments
BVPTU001	All	Copies the input to disk file

Sub-programs are used as an interface between VA Pac and the site's security system.

Program	Security system
BVPSECUR	RACF
BVPTSS	TOPSECRET Batch
BVPTSSC	TOPSECRET CICS

For RACF the BVPSECUR sub-program is supplied by SMP/E in the hlq.SBVPMBR8 PDS.

Refer to chapter 'Security system interface' in the 'VA Pacbase Installation Guide' for details on how to operate this extension.

Operation-parameter library

System parameters library: SY

Its required size is approximately 5 blocks of 6,080 bytes.

It is a PDS file which contains the input of the utilities used in the batch installation and operation procedures.

It includes the following:

The DELETE/DEFINE's for the VSAM files :

Members	Contents : DELETE/DEFINE of the file
DF$base.DA	DSMS data (DA)
DF$base.DC	VA Pac elements (DC)
DF$base.DJ	DSMS journal (DJ)
DF$base.DX	Cross-references (DX)
DFBVPDE	Error messages and HELP documentation (DE)
DF$base.DF	On-line DAF work file

The physical characteristics of the files are described in the subchapters which follow.

The information concerning the catalogue in use, the disks, the blocking factor, etc., is initialized according to the initial installation parameters and can be modified if need be by the DSMS manager.

The VERIFY's and LISTCAT's of the VSAM files:

- VERIFff (ff = DA, DC, DJ, DX, DE) members contain the VERIFY PACDff command for each DSMS file.
- The LI$base.DJ member contains the LISTCAT command for the Journal file (DJ).

IDCAMS utility's input:

- MAXKEY: Maximum record
- REPRO999: Copy request

Note: All modifications of file characteristics must be made in the System-Parameter Library.

'System' files

They make up the actual system. They are not affected by daily transactions, and they must be reloaded each time the system is reinstalled.

These are the LIBRARIES described in the previous subchapters:

- The library of on-line executable modules: SBVPMTR8,
- The library of batch executable modules: SBVPMBR8,
- The library of system parameters: BVPSY,

as well as the file containing the ERROR MESSAGES and the HELP DOCUMENTATION of the DSMS function (DE):

Characteristics	Value
Size	Approximately 30,000 records
Organization	VSAM-KSDS
Recsize	90
Key	17 (position 0)
DSN	&INDSV..BVPD

'User' files

These files contain the user information managed by DSMS.

The first four files contain the data directly managed by DSMS. They are:

- The DSMS Data file (DA)

Characteristics	Value
Organization	VSAM-KSDS
Recsize	80 minimum, 350 maximum
CI size	4,096
Key	40 (position 2)
DSN	&INDUV..&base.DA

- The Cross-References file (DX)

Characteristics	Value
Organization	VSAM-KSDS
Recsize	80
CI size	4,096
Key	50 (position 0)
DSN	&INDUV..&base.DX

- The VA Pac Elements file (DC)

Characteristics	Value
Organization	VSAM-KSDS
Recsize	50 minimum, 168 maximum
CI size	4,096
Key	31 (position 2)
DSN	&INDUV..&base.DC

- The DSMS Journal file (DJ)

Characteristics	Value
Organization	VSAM-RRDS
Recsize	180
CI size	4,096
DSN	&INDUV..&base.DJ

- The On-Line DAF work file (DF)

Characteristics	Value
Organization	VSAM-KSDS

Characteristics	Value
Recsize	100 min., 554 max.
CI size	4,096
Key	37 (position 2)
DSN	&INDUV..&base.DF

Three other sequential files constitute the DSMS backup. They are:

- The Backup file (BB)

Characteristics	Value
Organization	Sequential generation file
Recfm	Variable
Lrecl	354
Blksize	6,376
DSN	&INDUN..&base.BB(n)

- The Archived Journal file (BJ)

Characteristics	Value
Organization	Sequential generation file
Recsize	180
DSN	&INDUN..&base.BJ(n)

- The Deactivated Archived file (BQ)

Characteristics	Value
Organization	Sequential
Recsize	180
DSN	User-defined

Entry-point sources

The sources of user control sub-programs for the definitions of changes, events, sites, requests and layouts, and the DAF tables Dictionary (DAFDIC) can be downloaded through the VA Pac Support web page at

http://www.ibm.com/software/awdtools/vapacbase/support.html

Member	Contents
BVPCUAM	Online control on change definition
BVPCUEV	Online control on event definition
BVPCUMQ	Online control on layout definition
BVPCURQ	Online control on request definition
BVPCUSI	Online control on site definition
PDSCAM	Batch control on change definition
PDSCEV	Batch control on event definition
PDSCMQ	Batch control on layout definition
PDSCRQ	Batch control on request definition
PDSCSI	Batch control on site definition

Chapter 3. Environment

On-line environment

The monitor in use is CICS/ESA Version 4.1 or TS.

The SPOOL=YES option is necessary to run batch jobs (choice LVQ).

All User files are updated on-line and must be protected by the 'DYNAMIC TRANSACTION BACKOUT' (ROLLBACK) option.

The use of the CICS 'EMERGENCY RESTART' option is recommended.

The number of Strings, Indexbuffers and Databuffers allows the product to run correctly in most installations.

However, on the sites which contain very large volumes of data and a large number of users, these parameters must be adapted.

GENERAL INFORMATION - HOW THE SYSTEM RUNS

The general characteristics are:

- Only one transaction code is used to manage the DSMS data. It points to a 'monitor' program whose purpose is to link the different programs that make up the system. No screen is directly driven by this program.
- In case of a system ABEND, an 'ABEND MAP' is generated.
- Updates are serialized; in other words, the system protects the concurrent accesses to the database by placing all update transactions in a wait queue (ENQUEUE and DEQUEUE in the update cycle).

EXIT FROM THE DSMS TRANSACTION

A special program is automatically called when you exit the DSMS transaction (program code: BVPMOSO). This program performs an EXEC CICS RETURN, after displaying the "CURRENT DSMS CONVERSATION IS SAVED" if the exit is done by the '.12' option or by the corresponding function key (default: PF12).

If you wish the DSMS module not to branch off to CICS, you can replace this program with a user program containing for instance a call of user transaction.

DSMS sends a three-character COMMAREA to the program:

```
Positions 1-2: Blank or 12 (for a save of the
               conversation)
Position 3:   'F' or 'A' according to the installation
               language.
```

If you replace BVPMOSO with a program of your own, you can use the COMMAREA, and, if there is a '12' in positions 1-2, display the "CURRENT DSMS CONVERSATION IS SAVED" message (if an 'A' is in position 3), or the "LA CONVERSATION DSMS A ETE SAUVEGARDEE" message (if an 'F' is in position 3).

UPPERCASE AND LOWERCASE PROCESSING

The VA Pac-DSMS system has its own character management system:

- All codes entered in lowercase are automatically turned into uppercase,
- All clear names, labels and texts remain in lowercase.

In CICS ESA 3.2, the UCTRAN option can be controlled at transaction level.

VA Pac transactions must have the UCTRAN=NO option. (to use uppercase letters only, set the UCTRAN option to YES).

In CICS if the UCTRAN option of the terminal is active, it is automatically inhibited by VA Pac for the time of the connection, so that this control is available.

The installation process suggests 2 routine versions: BVPUCTR and BVPUCTX. The BVPUCTX routine (renamed BVPUCTR) can be used to inhibit the modification of the UCTRAN parameter in all the CICS releases, which is useful if you wish to always work in uppercase letters.

Batch environment

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

The amount of memory needed for the execution of batch procedures varies according to the size of the buffers allocated to the files they use.

Access method

The DSMS function manages its files using the indexed VSAM-KSDS access method without a secondary index, and the relative VSAM-RRDS access method.

All files are protected against concurrent write accesses (SHARE OPTION 2).

All batch procedures include DELETE/DEFINE steps to take care of file reloading. Therefore, the files do not need to include the REUSE option and, consequently, can be allocated using either the UNIQUE option or the SUB-ALLOCATION option.

Chapter 4. Batch procedures

Introduction

The batch processing associated with DSMS is divided into procedures. The following chapters describe each of these procedures that may be used and give details on its specific execution conditions.

For each procedure, you will find:

- A general presentation containing:
  - an introduction,
  - the execution condition(s),
  - the action(s) to be taken in case of abnormal execution,
- The description of user input, processing, results, and possible recommendations on use.
- A description of each step containing:
  - The files used (temporary and permanent),
  - The return codes that may be generated by each step.

Classification of procedures

There are various types of batch procedures.

DATABASE MANAGEMENT PROCEDURES:

- Initialization of DSMS files (DINI)
- Archiving of file update transactions (DARC)
- Restoration of files using the backup and archived files (DRST)
- Backup of files (DSAV)
- Reorganization of cross-references files (DREO).

UTILITY PROCEDURES:

- Extraction, from the VA Pac Journal, of the transactions which correspond to the modified VA Pac entities linked to the Changes (DEXP).
- Extraction, from the DSMS journal (DXBJ), of the transactions for the DUPT batch update.
- Printing of query results, and of table and keyword lists requests (DPRT).
- Extraction from DSMS of Events, Changes, Sites or Tables as batch transactions (DEXT).

- Batch update of DSMS files of Events, Changes, Sites or Tables (DUPT, DUPD).
- Pre-processing of DAF source files (DPDF).
- Renaming of Table, Site and Keyword codes (DREN).
- Printing of the installed programs list (DINS).

RETRIEVAL OF A DATABASE ON ANOTHER PLATFORM:

- Replacement of low-values with blanks (DLVB).

Abnormal execution

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 USER ABEND (code 12), accompanied by a message on the SYSOUT file.

When an abend occurs, the user 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
```

In most cases, examining the status and type of operation allows the user 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
	INFORMATION

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

If there is no message and if the type of ABEND is directly related to 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.

Chapter 5. DARC - Journal archiving

DARC - Introduction

The Journal Archiving procedure (DARC) backs up the Journal file (DJ) as a sequential file (BJ), and reinitializes it both logically and physically.

The new archived transactions do not overwrite the transactions previously archived; they are added to them.

The previously archived transactions can be deactivated, if requested.

Execution condition

The database must be closed to on-line use.

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'.

If the abnormal end occurs before the step which creates the Journal file (DJ), the procedure can be restarted as it is, after the problem has been solved.

If the abnormal end occurs during or after this step, the user input must be modified before a new execution of the procedure so as to specify a reinitialization request without a backup of the Journal file (already backed-up).

Caution

With systems using generation files (MVS for instance), the +1 version of the archived transaction file might have been cataloged even if the procedure ended abnormally. In this case, the procedure must be executed again with the -1 version of the archived transaction file (not the 0 version) as input.

DARC - Input / Processing / Results

USER INPUT

The DARC procedure includes an optional input for:

- deactivating the previously archived transactions that are now obsolete,
- indicating the absence of previously archived transactions during input,

- indicating the unavailability of the Data file (DA) during input,
- requesting a reinitialization of the transaction file only.

The structure of this input is as follows:

Pos.	Len.	Value	Meaning
2	1	'S'	Line code
3	4	nnnn	Session number
7	8	CCYYMMDD	OR date up to which the user requests deactivation
15	1	'I'	Absence of previously archived transactions
16	1	'D'	Data file (DA) unavailable
17	1	'J'	Re-initialization without archiving

The session number and the date are exclusive. They are ignored if the absence of previously archived transactions has been indicated.

The unavailability of the Data file is to be indicated only when this file has been physically deleted (see paragraph 'Recommendations').

The reinitialization request without a backup is necessary when the Journal file is physically destroyed.

Caution:

In this case, the previous archiving is not duplicated on the output archiving. When the cataloging is automatic, previous archiving may be lost if no uncataloging is performed.

In case of an error on one of the options, an error message is printed and the archiving is generated using the default options.

Recommendations

If there is no user input, this procedure can be executed only if the database is in a consistent state, and if the Journal file is correctly formatted.

When a database needs to be restored after a problem, some information in the database may be destroyed and neither the DARC nor the DRST procedures can then be executed.

In this case, AND IN THIS CASE ONLY, columns 15 to 17 of the user input must be used as follows:

- If the Data file (DA) is lost or has been flagged as 'inconsistent', a 'D' in column 16 means that the DARC procedure will not take the Data file (DA) into account. However, the DRST procedure must be executed afterwards, since under these conditions, the DARC procedure renders the database inconsistent.
- If the Journal file (DJ) is lost or destroyed, a 'J' must be entered in column 17. The DARC procedure formats an empty Journal file. The DRST procedure can then be executed.
- If the sequential Archived file (BJ) is lost or destroyed, an 'I' must be entered in column 15. The DARC procedure will format a new sequential archive file.

If one of these columns is accidentally set to its value, and the DARC procedure executed when the Data (DA) file is in a consistent state, the consequences are :

- 'I' in col. 15: The transactions previously archived are lost. All the transactions can be recovered by concatenating BJ(-1) and BJ(0) to obtain BJ(+1).
- 'D' in col. 16: The DARC procedure has to be re-run BEFORE any update. If it is done afterwards, the data is lost and a complete restoration must be executed.
- 'J' in col. 17 : The contents of the Journal file are lost and cannot be retrieved.

REPORT RESULTS

This procedure prints a report giving the number of archived update transactions and, if applicable, the number of records that have been deactivated.

GENERAL RESULTS

Once this procedure is executed, a sequential file containing all the archived transactions is produced.

The Journal file is re-initialized.

It is also possible to store in another file all update transactions that have been deactivated.

Note: This procedure does not increment the current session number of the database.

DARC - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Archiving of journal file: PDS300

This step executes the following processing:

- Updates the file of archived update transactions,
- Positions a flag in the Data file which represents the journal archiving,
- Writes the deactivated transactions onto a special file, if deactivation is requested by user input.

Code	Physical name	Type	Label
PACDMB	&&PAC7MB	Input	User transaction
PACDJB	&INDUN..&base.BJ(0)	Input	Already archived transactions
PACDDJ	&INDUV..&base.DJ	Input	Journal file to reinitialize
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDDA	&INDUV..&base.DA	Input Output	Data file
PACDBJ	&INDUN..&base.BJ(+1)	Output	Archived update transactions
PACDBQ	DUMMY	Output	Deactivated archived transactions (This file can be retrieved if necessary)
PACDRU		Report	Archiving review
SORTWK01 SORTWK02 SORTWK03		Sort	

Return codes::

- 0 : No error detected on the files.
- 8 : User Input error.
- 12 : Input-output error on a file.

Definition of the journal file: IDCAMS

This step executes a DELETE/DEFINE on the Journal file (DJ):

Code	Physical name	Type	Label
PACDDJ	&INDUV..&base.DJ	Output	Journal file

Re-initialization of the journal file: PDS320

This step executes the following:

- Creates a record in the Journal file
- Repositions the Data file flag.

Code	Physical name	Type	Label
PACDMB	&&PAC7MB	Input	User transaction
PACDDE	&INDSV..BVPDE	Input	Error-message file
PACDDA	&INDUV..&base.DA	Input Output	Data file
PACDDJ	&INDUV..&base.DJ	Output	Journal file to be reinitialized
PACDRU		Report	Reinitialization report

DARC - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*                  - ARCHIVAL OF DSMS JOURNAL -                     *
//*********************************************************************
//BVPDARC  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',         INDEX OF USER VSAM FILES
//  INDUN='$INDUN',         INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',         INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',        SORT LIBRARY
//*:  VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//  CYL=3,                  SORTWORK SPACE
//  DSCB='$DSCB',           DSCB MODEL FILE
//  OUT='$OUT',             OUTPUT CLASS
//  VOLS='SER=$VOLUN',      VOLUME OF ARCHIVED JOURNAL (BJ)
//  UNITS='$UNITUN',        ARCHIVED JOURNAL UNIT (DISK OR TAPE)
//  UWK=$UWK,               WORK UNIT
//  SPABJ='(TRK,(10,2),RLSE)'   SPACE OF JOURNAL (IF DISK)
//*********************************************************************
//* INPUT       : COMMAND FOR DEACTIVATION OF ARCHIVED TRANSACTIO
//* COL  2      : 'S'
//* COL  3 A  6 : SESSION NUMBER
//* COL  7 A 14 : DATE (CCYYMMDD)
//* COL  15     : ' ' PRESENCE OF INPUT ARCHIVED TRANSACTION FILE
//*             : 'I' ABSENCE OF INPUT ARCHIVED TRANSACTION FILE
//* COL  16     : ' ' PRESENCE OF DATA FILE (DA)
//*             : 'D' ABSENCE OF DATA FILE (DA)
//* COL  17     : ' ' ARCHIVAL AND REINITIALIZATION
//*             : 'J' REINITIALIZATION WITHOUT ARCHIVAL
//*
```

```
//* IN THE ABSENCE OF INPUT (OR ERROR ON A COMMAND PARAMETER),
//* NO DEACTIVATION WILL TAKE PLACE, HOWEVER ARCHIVAL AND
//* REINITIALIZATION WILL BE EXECUTED NORMALLY.
//*
//* TRANSACTIONS WHOSE SESSION (DATE) IS PRIOR OR EQUAL TO
//* THE SESSION (DATE) INDICATED ARE NOT KEPT. THEY ARE
//* RECOVERED IN THE FILE OF DEACTIVATED TRANSACTIONS.
//*                                                                    *
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDS300 EXEC PGM=BVPDS300,REGION=2048K
//*-----------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDJB   DD DSN=&INDUN..&BAS.0BJ(0),DISP=OLD
//PACDBJ   DD DSN=&INDUN..&BAS.0BJ(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//        SPACE=&SPABJ,
//        DCB=(&DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//PACDBQ   DD DUMMY,DCB=BLKSIZE=180
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELDEF EXEC PGM=IDCAMS,COND=(0,NE,PDS300)
//*--------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
```

```
//SYSIN DD DSN=&INDSN..BVPSY(DF&BAS.0DJ),DISP=SHR
//PDS320 EXEC PGM=BVPDS320,COND=(0,NE,PDS300)
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELBJ  EXEC PGM=IEFBR14,COND=(08,NE,PDS300)
//*---------------------------------------
//DDBJ     DD DSN=&INDUN..&BAS.0BJ(+1),DISP=(OLD,DELETE)
```

Chapter 6. DPRT - Printing of queries and output reports

DPRT - Introduction

The DPRT procedure performs all the printing operations for DSMS:

- Results of User Queries on Events, Changes and Sites, (this order must be respected)
- Printouts of Tables, Keywords, Queries and Reports.

See the DSMS Reference Manual for practical information on how to submit a DPRT execution in either batch or on-line mode.

Note: Printouts of Tables and Keywords can be submitted in batch mode only.

Technical information regarding the JOB Function (which enables you to submit the DPRT procedure in on-line mode) is given at the end of this chapter.

Execution conditions

None.

The Database can remain open to on-line processing.

Abnormal execution

Refer to Chapter THE BATCH PROCEDURES, Subchapter 'Abnormal Execution'.

DPRT - Input / Processing / Results

USER INPUT

A '*' line (required):

Col.	Len.	Value	Description
2	1	'*'	Line Code
3	8	uuuuuuuu	DSMS User Code
11	8	pppppppp	Password
19	3	ppp	Product Code
22	2	su	Subsidiary Code

Col.	Len.	Value	Description
24	1	l	Language Code

4 report types exist, 1 line per printout is necessary :

- Tables

Col.	Len.	Value	Description
02	03	Txx	Codes of the Txx table
07	02	C1	... with their labels in the language of the connected user (default option)
07	02	C2	... with all their labels
02	03	TUD	User codes with all their authorizations (TUG + TUP + TUS)

- Queries / Reports

Col.	Len.	Value	Description
02	04	X QC	Query on Changes
		X QE	Query on Events
		X QS	Query on Sites
02	04	X RC	Report on Changes
		X RE	Report on Events
		X RS	Report on Sites
06	06	xxxxxx	Query or Report code
12	08	uuuuuuuu	User code for Query or Report owner (default value: connected user code)
20	02	C1	Print of all description lines for the Query/Report type (default option)
		C2	Print of only useful Query/Report description lines

- Lists

Col.	Len.	Value	Description
02	03	LJQ	Control cards
02	04	LCQC	Query on Changes
		LCQE	Query on Events
		LCQS	Query on Sites
02	04	LCRC	Reports on Changes

Col.	Len.	Value	Description
		LCRE	Reports on Events
		LCRS	Reports on Sites
07	02	C1	Print of all description lines for the Query/Report type (default option)
		C2	Print of only useful Query/Report description lines
12	08	uuuuuuuu	User code for Query/Report owner

- Keywords

Col.	Len.	Value	Description
02	04	LAKC	Stand-alone Keywords of Changes
		LPKC	Principal keywords of Changes
		LGKC	All the keywords of Changes
06	01	l	Keywords language code (default: connected user language code)
02	04	LAKE	Stand-alone Native Keywords of Events
		LPKE	Principal Native Keywords of Events
		LGKE	All Native Keywords of Events
02	04	LAKT	Stand-alone Techn. Keywords of Events
		LPKT	All principal keywords of Events
		LGKT	All the keywords of Events

Print via user query (99 queries maximum):

Col.	Len.	Value	Description
2	1	'Q'	
3	1	'C'	For a query on Changes
		'E'	For a query on Events
		'S'	For a query on Sites
5	6	rrrrrr	Code of the user Query (required) 'Q' Entity used.
5	6	mmmmmm	Code of the Report (optional)
17	1	d	Delimiter (optional)
			-
			Parameters:
18	1	s	Symbol -

Col.	Len.	Value	Description
19	1	x	Separator -
20	54	.....	Parameter values -
			If optional fields have not been filled in, default values are used. They come from the definition lines of the user Query found in the Data- Base.

PRINTED OUTPUT

Two types of printed output are obtained:

- Results of user-defined Queries on Events, Changes and Sites.
- Standard printouts of Tables, Keywords, Queries and Reports.

Return code

Code	Description
0	OK with Queries
4	OK with tables, keywords, Queries/Reports print requests
8	OK with erroneous Queries or other requests
12	Fatal error
16	Sort error

DPRT - Description of steps

This procedure calls a unique program (PDSB) that acts as a flow monitor for the various programs, which are therefore sub-programs of this monitor.

The procedure includes the following steps:

Input recognition: PTU001

The input file is automatically formatted when QUERIES are submitted on-line.

Check of VSAM files: IDCAMS

Printing: PDSB

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file

Code	Physical name	Type	Label
PACDDC	&INDUV..&base.DC	Input	VA Pac elements file
PACDDE	&INDSV..BVPDE	Input	Error messages file
PACDMB	&&PACDMB	Input	User Queries
PACDKD		Workfile	Print requests
PACDKQ		Workfile	Queries
PACDID		Workfile	Temporary files
PACDQR		Workfile	Temporary files
PACDQJ		Workfile	Temporary files
PACDW1		Workfile	Temporary files
PACDW2		Workfile	Temporary files
PACDW3		Workfile	Temporary files
PACDW4		Workfile	Temporary files
PACDIA		Report	Flow report
PACDIB		Report	List of Queries and requests
PACDID		Report	Print of tables and keywords
PACDIQ		Report	Report of Query extractions
PACDQI		Report	Print of Query extractions
PACDRQ		Report	Print of Queries/Reports
PACDJQ		Report	Print of control cards
SORTWK01 SORTWK02 SORTWK03		Sort	

DPRT - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - PRINTOUTS AND QUERIES -                             *
//*********************************************************************
//BVPDPRT  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',         INDEX OF USER VSAM FILES
//  INDSV='$INDSV',         INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORTLIB
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SIZE
//  SPAMB='(TRK,(30,5),RLSE)', SPACE OF EXTRACTION COMMANDS
```

```
//  COPIES=1,              NUMBER OF REPORT COPIES
//  SPAWK='(CYL,(20,2))',  WORK FILE SPACE
//  OUT='$OUT',            OUTPUT CLASS
//  OUTL='$OUT',           QUERY OUTPUT CLASS
//  LNG=$LNG,    LANGUAGE OF MONITOR (E FOR ENGLISH F FOR FRENCH)
//  UWK=$UWK               WORK UNIT
//*****************************************************************
//* INPUT :
//* .. IDENTIFICATION LINE
//*  COL 02     : *
//*  COL 03     : DSMS USER CODE
//*  COL 11     : PASSWORD
//*  COL 19-21  : PRODUCT CODE
//*  COL 22-23  : SUBSIDIARY CODE
//*  COL 24     : LANGUAGE CODE
//*
//*  ..  EXTRACT COMMAND LINE(S)
//*      ----------------------
//*  COL 02-05  : TYPE OF EXTRACTION
//*  -- EXTRACTION BY USER QUERY :
//*  COL 05-10  : QUERY CODE
//*  COL 17     : DELIMITER                   <--- OPTIONAL
//*  COL 18     : SYMBOL                      <--- OPTIONAL
//*  COL 19     : SEPARATOR                   <--- OPTIONAL
//*  COL 20-73  : PARAMETERS VALUES           <--- OPTIONAL
//*  -- EXTRACTION OF QUERIES/LAYOUT :
//*  COL 06-11  : QUERY OR LAYOUT CODE
//*  COL 12-19  : OWNER OF THE QUERY/LAYOUT   <--- OPTIONAL
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN
//VERIFY EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDSB     EXEC PGM=BVPDSB&LNG,REGION=4096K
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
```

```
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDIA   DD SYSOUT=&OUTL
//PACDIB   DD SYSOUT=&OUTL
//PACDID   DD SYSOUT=&OUTL,COPIES=&COPIES
//PACDIQ   DD SYSOUT=&OUTL,COPIES=&COPIES
//PACDQI   DD SYSOUT=&OUTL,COPIES=&COPIES
//PACDRQ   DD SYSOUT=&OUTL,COPIES=&COPIES
//PACDJQ   DD SYSOUT=&OUTL,COPIES=&COPIES
//PACDQR   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6650
//PACDQJ   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6650
//PACDKD   DD UNIT=&UWK,SPACE=&SPAMB,DCB=BLKSIZE=6256
//PACDKQ   DD UNIT=&UWK,SPACE=&SPAMB,DCB=BLKSIZE=6160
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,DELETE,DELETE)
//PACDW1   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6160
//PACDW2   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6080
//PACDW3   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6375
//PACDW4   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6080
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 7. DRST - Database restoration

DRST - Introduction

The Database Restoration procedure (DRST) restores the files, using the sequential image produced by the Database Backup procedure (DSAV).

Archived transactions can also be retrieved once this procedure has been executed.

Execution conditions

The database must be closed to on-line processing.

These modifications must be made in the System Parameters library.

The procedure physically and logically re-initializes the Journal file which must have been saved previously by the DARC procedure.

These modifications must be made in the System Parameters library (SY).

Abnormal execution

Refer to Subchapter 'Abnormal execution' in Chapter 'The batch procedures'.

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

DEFINITION CONTROL SUB-PROGRAMS

These sub-programs (delivered as COBOL sources) are designed to add specific controls or initializations on the 5 DSMS definitions (Change, Event, Report, Request and Site).

Process:

When the screen is displayed for the first time, there is no access to the control sub-program.

When it is updated, the usual controls are first executed by the Definition screen, and then the sub-program is called. This sub-program will search for fatal errors (F40) and will send a message, whenever relevant, with an update lock, to the calling program which will just display the information.

If no error is found (or after the correction of errors, followed by the usual controls and a new call to the sub-program), the values entered are controlled again and a warning may be sent (F45). The user will then just have to press ENTER to take into account the value previoulsy entered.

Then, via a branching or a now call, the sub-program will be able to assign a new value to some input fields (F50).

Upon the return to the calling program, all the values (entered by the user or assigned by the sub-program) will be controlled again. This screen will then update the Database.

At the beginning, these sources only include 3 examples:

- 1 'WARNING'-type error
- 1 critical error
- 1 initialization.

Their linkage is made up of the displayed fields, the entered fields or some other fields directly or indirectly associated with the definition.

These sub-programs are called via tops indicated in the technical record of the DRST procedure.

There are 10 of them: 5 for on-line processing and 5 for batch processing.

Note: Errors are set using 'PR' (as in VAPAC) ; these fields must be set to 'W' or 'E'.

DRST - Input / Processing / Results

USER INPUT

The following chart lists the DRST procedure's input.

Pos.	Len.	Value	Meaning
2	1	'R'	Line code
3	1	'l'	Language code 'E' or 'F' (optional)
4	1		Journal inhibition flag
		'0'	No inhibition (default option)
		'1'	Inhibition
5	3	'REC'	Restoration and retrieval of archived transactions

Pos.	Len.	Value	Meaning
8	12		12-position table indicating the PFkeys assignment (default: 123456789ABC, but you may move or set to blank one or several values)
20	1		SECURITY SYSTEM INTERFACE
		' '	Retrieval of the previous value or no interface (for creation)
		'&'	Clear = Deactivation
		'R'	RACF
		'S'	TOPSECRET
22	1	'C'	Encryption of passwords
		'D'	Decryption of passwords
		' '	Unchanged passwords
			NOTE: it is not advised at all to request an encryption or decryption of passwords at the same time as the retrieval of archived transactions request (because the action is not performed on the journal).
25	1	'C'	Call of the sub-routine of additional controls for Change definition
		'&'	No call of sub-routine
26	1	'E'	Call of the sub-routine of additional controls for Event definition
		'&'	No call of sub-routine
27	1	'Q'	Call of the sub-routine of additional controls for Query definition
		'&'	No call of sub-routine
28	1	'R'	Call of the sub-routine of additional controls for Report definition
		'&'	No call of sub-routine
29	1	'S'	Call of the sub-routine of additional controls for Site definition
		'&'	No call of sub-routine

OUTPUT REPORT

This procedure prints a report listing the requested options, associated errors, the number of records restored in the database for each file, and the options memorized in the new database.

RESULT

Once this procedure is executed, the current session number is that of the sequential image or that of the most recent transaction, if the retrieval of archived transactions has been requested.

DRST - Description of steps

Check on Journal existence: IDCAMS

This step executes a LISTCAT on the Journal file (DJ). Its return codes are:

- 0 : The journal file exists
- OTHER : The journal file does not exist

Validation of journal contents: PDS380

This step is executed only when the Journal file exists. In this case, it checks that the journal has been archived.

Code	Physical name	Type	Label
PACDDJ	&INDUV..&base.DJ	Input	Journal file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDRU		Report	AJ file status report : It is printed if the journal file has not been archived.

Return codes:

- 0 : The Journal file was archived.
- 4 : The Journal file was not archived.

  (In this case, none of the DRST steps is executed).

Definition of files: IDCAMS

This step is executed only if the Journal file has been archived. It executes a DELETE/DEFINE on the Databae files.

Database restoration: PDS400

This step is executed only if the Journal file has been archived.

Code	Physical name	Type	Label
PACDBB	&INDUN..&base.BB(0)	Input	Backup of the files

Code	Physical name	Type	Label
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDMB	&&RESTMB	Input	User transactions
PACDDA	&INDUV..&base.DA	Output	Data file
PACDDC	&INDUV..&base.DC	Output	VA Pac elements file
PACDDJ	&INDUV..&base.DJ	Output	Journal file
PACDDX	&INDUV..&base.DX	Output	Cross-reference file
PACDMS	&&PACDMS	Output	Work file (2 records)
PACDRU		Report	Restoration report

Retrieval of archived journal: PDS450

This step is executed only when there are transactions to be retrieved. It does not cause a 'journalization' of transactions.

Code	Physical name	Type	Label
PACDMS	&&PACDMS	Input	Work file (2 records)
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDDA	&INDUV..&base.DA	Input Output	Data file
PACDDC	&INDUV..&base.DC	Input Output	VA Pac element file
PACDDX	&INDUV..&base.DX	Input Output	Cross-reference file
PACDBJ	&INDUN..&base.BJ(0)	Input	Archiving of the journal to retrieve
PACDRU		Report	Update report

DRST - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*              - LOADING-RESTORATION OF DSMS DATABASE -             *
//*********************************************************************
//BVPDRST  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
```

```
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK',            WORK UNIT
//  LNG=$LNG     LANGUAGE OF MONITOR (E FOR ENGLISH F FOR FRENCH)
//*************************************************************
//*  INPUT
//*   COL 02     : R
//*   COL 03     : INITIAL LANGUAGE CODE ( F=FRENCH, E=ENGLISH)
//*   COL 04     : 1   : INHIBITION OF TRANSACTION LOG
//*   COL 05     : MACHINE DATE FORMAT (I FOR MM/DD/YY)
//*              :                      (N FOR DD/MM/YY)
//*   COL 06-08  : REC : RETRIEVAL OF ARCHIVED TRANSACTIONS
//*   COL 09-20  : (NOT USED)
//*   COL 21     : SECURITY SYSTEM (R,S, ,&)
//*   COL 22     : USER CONTROL UNDER RACF (N, ,&)
//*   COL 23     : CRYPT/UNCRYPT OF PASSWORD (C,D, )
//*   COL 24-25  : (NOT USED)
//*   COL 26     : CALL OF SUB-PGM FOR CHANGES (C, ,&)
//*   COL 27     : CALL OF SUB-PGM FOR EVENTS (E, ,&)
//*   COL 28     : CALL OF SUB-PGM FOR QUERIES (Q, ,&)
//*   COL 29     : CALL OF SUB-PGM FOR LAYOUTS (R, ,&)
//*   COL 30     : CALL OF SUB-PGM FOR SITES (S, ,&)
//*
//*  IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (DJ) IS NOT
//*  REINITIALIZED, NO RESTORATION IS EXECUTED.
//*  IT IS THEREFORE NECESSARY TO EXECUTE THE DARC PROCEDURE FIRS
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&RESTMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//EXISDJ EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(LI&BAS.0DJ),DISP=SHR
//PDS380 EXEC PGM=BVPDS38&LNG,COND=(0,NE,EXISDJ)
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PDS380)
//*--------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(DF&BAS.0DA),DISP=SHR
//      DD DSN=&INDSN..BVPSY(DF&BAS.0DC),DISP=SHR
//      DD DSN=&INDSN..BVPSY(DF&BAS.0DJ),DISP=SHR
```

```
//      DD DSN=&INDSN..BVPSY(DF&BAS.0DX),DISP=SHR
//PDS400 EXEC PGM=BVPDS400,COND=(0,NE,PDS380)
//*--------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&RESTMB,DISP=(OLD,PASS)
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMS   DD DSN=&&PACDMS,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PACDRU   DD SYSOUT=&OUT
//PACDBB   DD DSN=&INDUN..&BAS.0BB(0),DISP=OLD
//SYSUDUMP DD SYSOUT=&OUT
//PDS450 EXEC PGM=BVPDS450,COND=((0,NE,PDS380),(0,NE,PDS400))
//*-------------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDBJ   DD DSN=&INDUN..&BAS.0BJ(0),DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMS   DD DSN=&&PACDMS,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 8. DSAV - Database backup

DSAV - Introduction

The purpose of the backup procedure (DSAV) is to convert the main files that make up DSMS into a BB sequential format.

The backed-up files are :

- The Data file (DA),
- The VA Pac Element file (DC),
- The Cross-reference file (DX).

Execution condition

The database must be closed to on-line processing in order to ensure its consistency during the execution of the DSAV procedure.

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'.

The main cause of an abend is that the database has not been closed to on-line use.

After correction, the procedure can be restarted as it is.

DSAV - Input / Processing / Results

USER INPUT

One optional line code.

Col.	Len.	Value	Designation
2	1	'O'	Line Code
3	3	'ENC'	Encryption of passwords
		'DEC'	Decryption of passwords
		' '	Unchanged passwords

REPORT RESULTS

Once the backup is executed, a report is printed. It includes the number of records saved in each file and the session number.

OUTPUT RESULT

The output is a single sequential file (BB) of variable length, containing the image of the three saved files.

If the database is in an inconsistent state after an abnormal end in the last update, the DSAV procedure is not executed.

Note: The DSAV procedure increments the current session number.

DSAV - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Database integrity check: PDSBAS

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDRS		Report	Validity report

Return code

This utility sends a return code 4 and causes an ABEND if the Database is invalid.

Database backup: PDS500

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input Output	Data file
PACDDC	&INDUV..&base.DC	Input	VA Pac element file
PACDDX	&INDUV..&base.DX	Input	Cross-reference file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDMB	&&DSAVMB	Input	User transactions
PACDBB	&INDUN..&base.BB(+1)	Output	Sequential image of files
PACDRU		Report	Backup report

DSAV - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*              - BACKUP OF THE DSMS DATABASE -                      *
//*********************************************************************
//BVPDSAV  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',         INDEX OF USER VSAM FILES
//  INDUN='$INDUN',         INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',         INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  DSCB='$DSCB',           DSCB MODEL FILE
//  OUT='$OUT',             OUTPUT CLASS
//  VOLS='SER=$VOLUN',      VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',        BACKUP UNIT (DISK OR TAPE)
//  UWK='$UWK',             WORK UNIT
//  SPABB='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//*************************************************************
//* INPUT
//*  COL 02     : 'O'
//*  COL 03-05  : CRYPT, UNCRYPT (ENC,DEC,   )
//*
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&DSAVMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDX),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDSBAS EXEC PGM=BVPDSBAS
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
```

```
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDS500 EXEC PGM=BVPDS500,COND=(0,NE,PDSBAS)
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMB   DD DSN=&&DSAVMB,DISP=(OLD,PASS)
//PACDBB   DD DSN=&INDUN..&BAS.0BB(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPABB,
//            DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELBB  EXEC PGM=IEFBR14,COND=(08,NE,PDS500)
//*---------------------------------------
//DDBB     DD DSN=&INDUN..&BAS.0BB(+1),DISP=(OLD,DELETE)
```

Chapter 9. DREO - Reorganization of cross-reference file

DREO - Introduction

The Cross-Reference Reorganization procedure (DREO) rebuilds a sequential image of the database using another sequential image as a starting point. The resulting file is used as input to the Restoration (DRST) procedure.

The operating principle of this procedure is to rebuild the cross-references associated with the data from the 'image' of this data.

Execution conditions

The database can remain open during reorganization since the procedure operates on sequential images of the database (backups).

The updates executed after the file backing up used for reorganization, can be retrieved during the restoration of the reorganized database.

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'.

In case of an abnormal end, the procedure must be restarted from the beginning.

DREO - Input / Processing / Results

USER INPUT

Three different types of user input can be entered, but only one line of each type can be created.

The format of this input is given in the table below.

Pos.	Len.	Value	Meaning
1	1	Not Used	
2	1	'P'	Deletion of Products
	1	'S'	Deletion of Subsidiaries
	1	'X'	Deletion of Product/Subsidiary
3	60	Product code	(20 x 3 char.) if Col.2 = 'P'

Pos.	Len.	Value	Meaning
	60	Subsid. code	(30 x 2 char.) if Col.2 = 'S'
	60	Prod./ Subsid.	(12 x 5 char.) if Col.2 = 'X'

REPORT

This procedure prints messages stating inconsistencies found in the Data file.

RESULT

The result of this procedure is a reorganized sequential image of the DSMS database, used as input to the Restoration (DRST) procedure.

DREO - Description of steps

Input recognition: PTU001

Building of indexes (not keywords): PDSR10

Code	Physical name	Type	Label
CARTE	&&PACDMB	Input	Transactions
PACDBB	&INDUN..&base.BB(0)	Input	DSMS database backup
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDW1	&&W1	Workfile	VA Pac elements and data
PACDW2	&&W2	Workfile	Keywords and keyword references
PACDW3	&&W3	Workfile	Cross-references (not keywords)
PACDRH		Report	Inconsistencies in DSMS data
PACDRK		Report	Reorganization report
SORTWK01 SORTWK02 SORTWK03		Sort	

Building of keyword indexes: PDSR20

Code	Physical name	Type	Label
PACDW2	&&W2	Workfile	Keywords and keyword references
PACDW4	&&W4	Workfile	Keywords

Code	Physical name	Type	Label
PACDW5	&&W5	Workfile	Keyword references
SORTWK01 SORTWK02 SORTWK03		Sort	

Merge of indexes: PDSR30

Code	Physical name	Type	Label
PACDW3	&&W3	Workfile	Cross-references (except keywords)
PACDW5	&&W5	Workfile	Keyword references
PACDW6	&&W6	Workfile	Keyword references
SORTWK01 SORTWK02 SORTWK03		Sort	

General merge for backup: PDSR40

Code	Physical name	Type	Label
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDW1	&&W1	Workfile	VA Pac elements and data
PACDW4	&&W4	Workfile	Keywords
PACDW6	&&W6	Workfile	Keyword references
PACDBB	&INDUN..&base.BB(+1)	Output	Reorganized DSMS database backup
PACDRR		Report	Reorganization report
SORTWK01 SORTWK02 SORTWK03		Sort	

DREO - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*              - DSMS REORGANIZATION -                              *
//*********************************************************************
//BVPDREO  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
```

```
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  CYL='(4,1)',           SORTWORK SPACE
//  DSCB='$DSCB',          DSCB MODEL FILE

//  OUT='$OUT',            OUTPUT CLASS
//  UWK=$UWK,              WORK UNIT
//  SPADA='(TRK,(60,5),RLSE)',  WORK SPACE (DA + DC)
//  SPADX='(TRK,(60,5),RLSE)',  WORK SPACE (DX)
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//*************************************************************
//* OPTIONAL INPUT
//*  COL 02     : DELETION OF PRODUCTS, SUBSIDIARIES OR
//*               PRODUCT/SUBSIDIARY ENVIRONMENT (P,S,X)
//*  COL 03-62  : 20 PRODUCTS, 30 SUBSIDIARIES OR
//*               12 PRODUCT/SUBSIDIARY ENVT
//*
//*************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//PDSR10  EXEC PGM=BVPDSR10
//*-----------------------
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//PACDBB    DD DSN=&INDUN..&BAS.0BB(0),DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRH    DD SYSOUT=&OUT
//PACDRK    DD SYSOUT=&OUT
//CARTE     DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDW1    DD DSN=&&W1,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=VB,BLKSIZE=6022,LRECL=354)
//PACDW2    DD DSN=&&W2,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADX,
//          DCB=(RECFM=FB,BLKSIZE=6240,LRECL=120)
//PACDW3    DD DSN=&&W3,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=FB,BLKSIZE=6400,LRECL=80)
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP  DD SYSOUT=&OUT
//PDSR20  EXEC PGM=BVPDSR20
//*-----------------------
```

```
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//PACDW2    DD DSN=&&W2,DISP=(OLD,PASS)
//PACDW4    DD DSN=&&W4,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADX,
//          DCB=(RECFM=FB,BLKSIZE=6120,LRECL=340)
//PACDW5    DD DSN=&&W5,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=FB,BLKSIZE=6400,LRECL=80)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//SYSUDUMP  DD SYSOUT=&OUT
//PDSR30  EXEC PGM=BVPDSR30
//*-----------------------
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//PACDW3    DD DSN=&&W3,DISP=(OLD,PASS)
//PACDW5    DD DSN=&&W5,DISP=(OLD,PASS)
//PACDW6    DD DSN=&&W6,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=FB,BLKSIZE=6400,LRECL=80)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//SYSUDUMP  DD SYSOUT=&OUT
//PDSR40  EXEC PGM=BVPDSR40
//*-----------------------
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDBB    DD DSN=&INDUN..&BAS.0BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDRR    DD SYSOUT=&OUT
//PACDW1    DD DSN=&&W1,DISP=(OLD,PASS)
//PACDW4    DD DSN=&&W4,DISP=(OLD,PASS)
//PACDW6    DD DSN=&&W6,DISP=(OLD,PASS)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUX    DD SYSOUT=&OUT
//SYSOUT    DD SYSOUT=&OUT
//SYSUDUMP  DD SYSOUT=&OUT
//DEL12  EXEC PGM=IEFBR14,COND=(12,NE,PDSR40)
//*--------------------------------------
//DDBB     DD DSN=&INDUN..&BAS.0BB(+1),DISP=(OLD,DELETE)
```

Chapter 10. DEXP - Extraction from VA Pac archived journal

DEXP - Introduction

The Archived Journal Extraction procedure (DEXP) extracts transactions associated to Changes from the VA Pac Archived Journal file, and formats them in order to update, in the DSMS Database, the modified elements corresponding to each Change.

Execution conditions

None.

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'.

If an abnormal end occurs, the procedure can be restarted as it is, after the problem has been solved.

DEXP - Input / Processing / Results

USER INPUT

One '*'-line is required:

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

One extraction line is also required:

Pos.	Len.	Value	Meaning
2	1	'J'	Line code (required)
			THE FOLLOWING FIELDS ARE OPTIONAL :
3	1	' '	List of selected transactions
		'N'	No list

Pos.	Len.	Value	Meaning
4	24		Selection in the VA Pac Database:
4	4	nnnn	Session number, begin. of selection
8	4	pppp	Session number, end of selection
			--> Selection on session(s) prohibits selection on date(s)
12	8	CCYYMMDD	Starting date for selection
		'TODAY'	Starting date = current date
20	8	CCYYMMDD	Ending date for selection
		'TODAY'	Ending date = current date (default value if starting date ='today')
28	1		Version of selected transactions
		' '	Selection of all sessions
		'T'	Selection of frozen session
		'Z'	Selection of current session
29	3	ppp	Product code
32	4	xxxx	VA Pac Database logical code
36	3	lll	Code of selected library
39	16		Type of selected entities
55	1	' '	Extraction of transactions made under change 999999
		'N'	No extraction of 999999-change transactions
56	1	' '	Printing of duplicate transactions for the same VA Pac entity
		'N'	No printing of duplicate transactions
57	6	nnnnnn	Change number

REPORT

Extraction report showing the list of formatted transactions.

RESULT

A DSMS database update transaction file to be used as input to the DUPT procedure.

DEXP - Description of steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Transaction extraction and formatting: PDS600

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
PAC7PJ	&PAC7PJ	Input	VA Pac archived journal
PACDMB	&&EXPJMB	Input	User transactions
PACDMV	&&PACDMV	Output	DUPT update transaction file
PACDRU		Report	Report on selection request
SORTWK01 SORTWK02 SORTWK03		Sort	

Return codes:

- 0 : No error and no list requested
- 4 : No error and printout of the transactions list
- 8 : Error on the user line code or parameter line
- 12 : I/O error on a file

Printing of DSMS update transactions: PDS610

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDMV	&&PACDMV	Input	DSMS update transactions file
PACDRU		Report	List of update transactions

Return codes:

- 0 : No error
- 12 : I/O error on a file

DEXP - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*              - EXTRACTION-UPDATE OF DSMS DATABASE -               *
//*********************************************************************
//BVPDEXP  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SPACE
//  SPAMV='(TRK,(1,1))',   SPACE OF EXTRACTED TRANSACTIONS
//  PAC7PJ='NULLFILE',     DSN OF ARCHIVED TRANSACTIONS FILE
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK'             WORK UNIT
//*****************************************************************
//*   FORMAT OF TRANSACTIONS AT INPUT :
//*   .. A DSMS USER AND PASSWORD LINE
//*  COL 02     : *
//*  COL 03     : DSMS USER CODE
//*  COL 11     : PASSWORD
//*   .. COMMAND LINE(S) FOR EXTRACTION
//*  COL 02     : J
//*  COL 03     : ' ' SELECTED TRANSACTIONS LIST
//*             : 'N' NO LIST OF SELECTED TRANSACTIONS
//*  COL 04-07  : STARTING SESSION NUMBER
//*  COL 08-11  : ENDING SESSION NUMBER
//*  COL 12-19  : STARTING DATE (CCYYMMDD)
//*  COL 20-27  : ENDING DATE (CCYYMMDD)
//*  COL 28     : VERSION OF SELECTED TRANSACTIONS
//*             : ' ' ALL SESSIONS
//*             : 'T' FROZEN SESSIONS
//*             : 'Z' CURRENT SESSION                                 *
//*  COL 29-31  : PRODUCT CODE
//*  COL 32-35  : INTERNAL PACBASE DATABASE CODE
//*  COL 36-38  : LIBRARY CODE
//*  COL 39-54  : TYPE OF ENTITIES TO BE SELECTED
//*  COL 55     : EXTRACT OF TRANSAC. FOR CHANGE 999999 ( ,N)
//*  COL 56     : PRINTING OF ALL TRANSACTIONS          ( ,N)
//*  COL 57-62  : CHANGE NUMBER
//*                                                                   *
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
```

```
//*---------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDS600 EXEC PGM=BVPDS600
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDMV   DD DSN=&&PACDMV,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAMV,
//         DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PAC7PJ   DD DSN=&PAC7PJ,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDS610 EXEC PGM=BVPDS610,COND=(4,NE,PDS600)
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMV   DD DSN=&&PACDMV,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 11. DEXT - Extraction of entities

DEXT - Introduction

The Entity Extraction procedure (DEXT) extracts all DSMS entities and formats them into batch transactions to be used as input to the DSMS Database Update procedure (DUPT).

Principle

In order to select the extraction of Changes, Events or Sites, the procedure uses Queries (″Q″ entities) that must have been previously defined in the DSMS Database. These three types of extraction must be requested in the above order.

The Query code should also be specified in the extraction request (see 'User Input').

The screen Report (″R″ entity) associated with the Query used for the extraction does not interfere in the extraction.

Execution conditions

None.

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'.

If an abnormal end occurs, the procedure can be restarted as it is after the problem has been solved.

DEXT - Input / Processing / Results

USER INPUT

One '*'-line is required:

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

Pos.	Len.	Value	Meaning
19	3	ppp	Product code
22	2	su	Subsidiary code
24	1	l	Language code

Four types of extractions are available. One line per request is necessary:

Pos.	Len.	Value	Meaning
02	03	'PL'	Locking of databases
02	03	Txx	Codes of the Txx table (all tables except TRA)

- Queries / Reports:

Pos.	Len.	Value	Meaning
02	04	X QC	Query on Changes
		X QE	Query on Events
		X QS	Query on Sites
02	04	X RC	Report on Changes
		X RE	Report on Events
		X RS	Report on Sites
12	08	uuuuuuuu	Owner of the Query or Report (Default=logged-in user)

- Lists

Pos.	Len.	Value	Meaning
02	04	LCQC	Queries on Changes
		LCQE	Queries on Events
		LCQS	Queries on Sites
02	04	LCRC	Reports on Changes
		LCRE	Reports on Events
		LCRS	Reports on Sites
12	08	uuuuuuuu	Owner of Queries or Reports

- Keywords

Pos.	Len.	Value	Meaning
02	04	LAKC	Stand-alone keywords of Changes
		LGKC	All Changes' Keywords

Pos.	Len.	Value	Meaning
06	01	l	Language code of Keywords (Default=Language of logged-in user)
02	04	LAKE	Native stand-alone Keywords of Events
		LGKE	All Events' Keywords
02	04	LAKT	Techn. stand-alone Keywords of Events
		LGKT	All Keywords

Extraction via user request (99 requests maximum)

Pos.	Len.	Value	Meaning
2	1	'Q'	
3	1	'C'	For a query on Changes
		'E'	For a query on Events
		'S'	For a query on Sites
5	6	rrrrrr	User Query code (required) - 'Q' Entity use
5	6	mmmmmm	Report code (optional)
17	1	d	Delimiter (optional)
			Parameter settings:
18	1	s	Symbol -
19	1	x	Separator -
20	54	.....	Parameter values -
			If some optional fields were not completed, default values will be used. They come from the User Query's definition lines found in the Database.

PRINTED OUTPUT

Extraction report showing the number of extracted transactions.

RESULT

DSMS database update transactions to be used as input to the DUPT procedure.

This procedure displays a general return code:

Code	Description
0	OK
8	Error on the user line code or on a command line
12	I/O error on a file or Inconsistency of DSMS database
16	Sort error

DEXT - Description of steps

This procedure calls a single program (PDSEX) that acts as a flow monitor for all programs, which are then considered as its sub-programs.

The procedure includes the following steps:

Input recognition: PTU001

Check of VSAM files: IDCAMS

Extractions: PDSEX

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDC	&INDUV..&base.DC	Input	VA Pac element file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDMB	&&PACDMB	Input	Extraction requests
PACDKQ		Workfile	Queries
PACDIM	&&PACDIM	Output	Extracted batch transactions
PACDIA		Report	Flow report
PACDRU		Report	Extraction request report
SORTWK01 SORTWK02 SORTWK03		Sort	
PACDW0		Workfile	Temporary file
PACDW1		Workfile	Temporary file
PACDW2		Workfile	Temporary file
PACDW3		Workfile	Temporary file
PACDW4		Workfile	Temporary file
PACDW5		Workfile	Temporary file

Code	Physical name	Type	Label
PACDWI		Workfile	Temporary file

DEXT - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - EXTRACTION OF BATCH TRANSACTIONS FOR DUPT -         *
//*********************************************************************
//BVPDEXT  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',         INDEX OF USER VSAM FILES
//  INDSV='$INDSV',         INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',        SORTLIB
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                  SORTWORK SIZE
//  SPAIM='(TRK,(150,15),RLSE)', SPACE OF EXTRACTED TRANSACTIONS
//  SPAMB='(TRK,(150,15))', SPACE OF EXTRACTION COMMANDS
//  SPAWK='(CYL,(20,2))',  WORK FILE SPACE
//  OUT='$OUT',             UTILITIES OUTPUT CLASS
//  OUTL='$OUT',            OUTPUT CLASS OF REPORTS
//  LNG=$LNG,    LANGUAGE OF MONITOR (E FOR ENGLISH F FOR FRENCH)
//  UWK=$UWK                WORK UNIT
//*****************************************************************
//* INPUT :
//* .. IDENTIFICATION LINE
//*  COL 02     : *
//*  COL 03     : DSMS USER CODE
//*  COL 11     : PASSWORD
//*  COL 19-21  : PRODUCT CODE
//*  COL 22-23  : SUBSIDIARY CODE
//*  COL 24     : LANGUAGE CODE
//*
//*  ..  EXTRACT COMMAND LINE(S)
//*      ----------------------
//*  COL 02-05  : TYPE OF EXTRACTION
//*  -- EXTRACTION BY USER QUERY :
//*  COL 05-10  : QUERY CODE
//*  COL 17     : DELIMITER                  <--- OPTIONAL
//*  COL 18     : SYMBOL                     <--- OPTIONAL
//*  COL 19     : SEPARATOR                  <--- OPTIONAL
//*  COL 20-73  : PARAMETERS VALUES          <--- OPTIONAL
//*  -- EXTRACTION OF QUERIES/LAYOUT :
//*  COL 06-11  : QUERY OR LAYOUT CODE
//*  COL 12-19  : OWNER OF THE QUERY/LAYOUT   <--- OPTIONAL
//*                                                                   *
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
```

```
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDSEX    EXEC PGM=BVPDSEX&LNG
//*-------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDIA   DD SYSOUT=&OUTL
//PACDKQ   DD UNIT=&UWK,SPACE=&SPAMB,DCB=BLKSIZE=6160
//PACDIM   DD DSN=&&PACDIM,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAIM,
//            DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PACDRU   DD SYSOUT=&OUTL
//PACDW0   DD UNIT=&UWK,SPACE=&SPAMB,DCB=BLKSIZE=6160
//PACDW1   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6160
//PACDW2   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6080
//PACDW3   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6375
//PACDW4   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6080
//PACDW5   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6375
//PACDWI   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6187
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 12. DUPT - Batch update of entities

DUPT - Introduction

The Batch Update of Entities procedure (DUPT) updates the DSMS entities with transactions from the DEXT, DEXP and/or DXBJ procedures.

Transactions can also be entered directly in a file, using an editor. For a complete description of the batch transactions, see the 'BATCH TRANSACTIONS STRUCTURE', in the appendix of the DSMS Reference Manual.

Execution condition

The DSMS files must be closed to on-line use.

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'. If an abend occurs, and after the problem is solved,

- if a batch rollback is executed on the platform, you can restart the procedure as it is,
- otherwise, you must first execute a restoration with the retrieval of archived transactions.

CAUTION:

This procedure performs a GLOBAL update. Therefore, make sure that all the data fields have been filled in. The data fields that are not filled in will automatically be set to blank.

The Change, Event and Site definition screens require two update lines, and both lines must be filled.

DSMS automatically allocates numbers to Events or Changes when they are created. However, for its creation, an Event or Change must be allocated a temporary number. For example, to create a Change: C000001, where 000001 is the temporary number that DSMS will automatically replace with a unique number.

You must set the action code to 'C', since the system does not provide for implicit creation.

Several Changes or Events can be created simultaneously. In this case, each Change or Event being created must be allocated a different temporary number. For example, to create 3 Changes simultaneously: C000001, C000002 and C000003.

Note: Each transaction stream can only contain 2,520 changes and 2,520 events maximum (internal limit of the program).

DUPT - Input / Processing / Results

USER INPUT

- One Parameter line (optional).
- One Identification line per Product/Subsidiary concerned by the updates (required).
- Update transactions extracted and formatted by the DEXT, DEXP or DXBJ procedures.
- The user must add at least one identification line in front of update transactions.

Parameter line (optional)

Col	Len	Value	Description
2	1	$	LINE CODE
3	1		UPDATE MODE / SORT ORDER
			Defines the update or processing mode to be used by ALL userids for this execution of the DSMS batch procedure.
		A	NORMAL UPDATE MODE
			- Transactions sorted in ascending order before any update is applied (i.e. entity definitions are processed before sub-screens.)
			- Update mode specified for each sign-on record.
		D	DELETE MODE
			- Transactions sorted in descending order before any update is applied.
			- All transactions processed as Deletions - Action Code D'.
			- Sign-on records must specify 'NORMAL' mode - all other modes are considered as errors.
4	1		REPORT FORMAT INDICATOR
		1	SINGLE REPORT FORMAT

Col	Len	Value	Description
			- One 'END OF REPORT' line is produced.
			- The transaction 'INPUT NUMBER' is simply incremented by one for each transaction.
		2	SIGN-ON / USERID FORMAT 1
			- An 'END OF REPORT' line is produced for each userid / sign-on record.
			- The transaction 'INPUT NUMBER' is reset to one for each sign-on record. The sign-on record will appear as transaction number one.
		3	SIGN-ON / USERID FORMAT 2
			- An 'END OF REPORT' line is produced for each userid / sign-on record.
			- The transaction 'INPUT NUMBER' is reset to zero for each sign-on record. The sign-on record will appear as transaction number zero.

If the parameter line is not entered, ' $A1' is assumed.

Sign-on line format (required)

Col	Len	Value	Description
1	1		ACTION CODE / UPDATE MODE
			This field defines the update mode processing to be used for this userid.
		blank	NORMAL UPDATE MODE.
			- Works like DSMS on-line.
			- If an Event or Change is created, all following sub-screen transactions will be modified accordingly.
		V	VERSION CONTROL MODE.
			- All batch transactions will be processed with Action Code 'C' (create).
			- The external reference fields on Event and Change Definitions will be filled in.
			- The associated change fields on Event Definitions will be converted to the 'new' Change Number - the number assigned when the Change is created.
		R	REORGANIZATION MODE.

Col	Len	Value	Description
			The same as 'V' except that the external reference fields' content will not be altered.
2	1	*	SIGN-ON RECORD CODE
3	8	...	DSMS USER
11	8	...	DSMS uSER PASSWORD
19	3	ppp	PRODUCT CODE to which updates apply.
22	2	ss	SUBSIDIARY CODE to which batch updates apply.
24	1	blank	Unused
25	9		EXTERNAL REFERENCE VALUES
			The value of the next three fields is used to create Event and Change external references if the update mode is 'V'.
25	4	dddA	- DSMS external Database code
29	3	ppp	- DSMS external Product code
32	2	ss	- DSMS external Subsidiary code
34	1		BLANK LINE AFTER ERROR INDICATOR
		blank	A blank line is printed after each error message on the report.
		N	Blank lines are not printed after error messages on the report.
35	1		REPORT PAGE BREAK INDICATOR
		blank	A new page begins only when the number of lines per page exceeds the maximum number
		T	A page skip for each new type of transaction
		E	A new page for each transaction type of each entity
36	1		TRANSACTION SORT INDICATOR
		blank	The transactions are sorted by type before they are processed.
		N	The transactions are processed in their order of arrival.

REPORT

The printout generated by this procedure is an update report, with comments about irregularities or inconsistencies encountered during execution.

RESULT

The result of this procedure is:

- A DSMS database ready for on-line or batch processing,
- A Journal file of the transactions which have modified the database, if 'journalization' was not inhibited during the last restoration.

Note: This procedure increments the session number if it is the first access to the database for the current day.

DUPT - Description of steps

Database integrity check: PDSBAS

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDRS		Report	Validity report

Return code

This utility sends a return code 4 and causes an ABEND if the Database is invalid.

Check of VSAM files: IDCAMS

Update of the DSMS database: PDSUP0

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input Output	Data file
PACDDC	&INDUV..&base.DC	Input Output	VA Pac element file
PACDDX	&INDUV..&base.DX	Input Output	Cross-reference file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDIM	&INPUT	Input	Update transactions obtained via the DEXP procedure
PACDDJ	&INDUV..&base.DJ	Output	Journal file
PACDRP		Report	Update review
SORTWK01 SORTWK02 SORTWK03		Sort	

Return codes

- 0: No error
- 8: Error on the user line code or parameter line
- 12: I/O error on a file

DUPT - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - UPDATE OF THE DSMS DATABASE -                       *
//*********************************************************************
//BVPDUPT  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//  UWK='$UWK',            UNIT OF WORK FILES
//  CYL=3,                 SORTWORK SPACE
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  INPUT='NULLFILE'       INPUT TRANSACTIONS DSN
//*****************************************************************
//* INPUT :
//* .. PARAMETERS LINE (OPTIONAL)
//*  COL 02     : $
//*  COL 03     : UPDATE MODE (A,D)
//*  COL 04     : REPORT FORMAT INDICATOR (1,2,3)
//* .. IDENTIFICATTION LINE (MANDATORY)
//*  COL 01     : ACTION CODE / UPDATE MODE (V,R, )
//*  COL 02     : *
//*  COL 03-10  : USER CODE
//*  COL 11-18  : PASSWORD
//*  COL 19-21  : PRODUCT CODE
//*  COL 22-23  : SUBSIDIARY CODE
//*  COL 24     : (NOT USED)
//*  COL 25-31  : EXTERNAL REFERENCE VALUE (DATABASE, PRODUCT,
//*               SUBSIDIARY)
//*  COL 34     : BLANK LINE AFTER ERROR ( ,N)
//*  COL 35     : REPORT PAGE BREAK INDICATOR ( ,T,E)
//*  COL 36     : TRANSACTION SORT INDICATOR ( ,N)
//*
//* .. COMMAND LINES
//*********************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
```

```
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDX),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDSBAS EXEC PGM=BVPDSBAS
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDSUP0 EXEC PGM=BVPDSUP0,REGION=4096K,COND=(0,NE,PDSBAS)
//*-----------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDIM   DD DSN=&INPUT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRP   DD SYSOUT=&OUT
//SYSIN    DD DUMMY,DCB=BLKSIZE=80
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
```

Chapter 13. DINI - File initialization

DINI - Introduction

The DINI procedure initializes the files needed for the installation of a new DSMS database.

It provides an initial backup of the DSMS files, which must be loaded by the Database Restoration (DRST) procedure.

Execution conditions

None.

However, the parameters of the new DSMS database must have been previously defined, and must be different from the parameters in any other existing DSMS database on the site.

The initial allocation and loading of DSMS components must have been executed (see the Installation Process).

Abnormal execution

Refer to Subchapter 'Abnormal Execution' in Chapter 'The batch procedures'.

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

DINI - Input / Processing / Results

USER INPUT

The structure of the input is as follows:

Pos.	Len.	Value	Meaning
2	1	'I'	Line code
3	1	'l'	Initial language code (E by default: English)
4	1		This field is ONLY used with DOS/VSE
		'I'	Default option for all hardware
		'N'	If CURRENT-DATE = DD/MM/YY in DOS/VSE

REPORT

This procedure prints a report listing the memorized options and the number of initial records of the DSMS database files.

RESULT

The result is an initial backup including:

- an initial user, whose userid is '********' and whose password is '********' (See the paragraph that follows: INITIAL CONNECTION.)
- a record in the Language Table corresponding to the language code indicated in the user input.

IMPORTANT:

INITIAL CONNECTION:

The Database Restoration (DRST) procedure must be executed after the DINI procedure. After a successful execution of the DRST procedure, the DSMS database is installed.

Check that the on-line access to the new DSMS database is operational.

The initial connection to the DSMS database is executed as follows:

- Access the DSMS database.
- On the Sign-on screen, enter '********' as the user code and '********' as the password, then press the ENTER key.
- Among the choices listed on the menu, only those marked with a '*' may be accessed. They correspond to the Tables which must be updated for a proper operation of DSMS. The information must be entered in the Tables in the following order:
  - In the Languages Table (CHOICE: 'TLA'): the codes and labels of the languages used.
  - In the Products Table (CHOICE: 'TPR'): the product codes and labels.
  - In the Subsidiaries Table (CHOICE: 'TSU'): the subsidiary codes and labels.
  - In the User Parameters Tables (CHOICES: 'TUD', 'TUG', 'TUP' and 'TUS'): user codes and authorizations.

(For more details on the management of these tables, see the DSMS Reference Manual).

The '********' user code cannot be deleted; after the User Parameters Tables are updated, the DSMS Database Manager can change its password in order to prevent other users from using this code.

DINI - Description of steps

Input recognition: PTU001

Initial database backup: PDSINI

Code	Physical name	Type	Label
PACDMB	&&DINIMB	Input	Initialization transaction
PACDDE	&INDSV..BVPDE	Input	Error messages
PACDBB	&INDUN..&base.BB(+1)	Output	Sequential image of files
PACDRU		Report	Backup report

DINI - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - INITIALIZATION OF THE DSMS DATABASE -               *
//*********************************************************************
//BVPDINI  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDUN='$INDUN',     INDEX OF USER   NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  DSCB='$DSCB',       DSCB MODEL FILE
//  OUT='$OUT',         OUTPUT CLASS
//  VOLS='SER=$VOLUN',  VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',    BACKUP UNIT (DISK OR TAPE)
//  UWK='$UWK',         WORKFILE UNIT
//  SPABB='(TRK,(10,2),RLSE)' SPACE OF BACKUP (IF DISK)
//*****************************************************************
//*  INPUT
//*  COL  2     : I
//*  COL  3     : INITIAL LANGUAGE CODE ( F=FRENCH, E=ENGLISH)
//*  COL  4     : MACHINE DATE FORMAT (I FOR MM/DD/YY)
//*             :                     (N FOR DD/MM/YY)
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&DINIMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//PDSINI EXEC PGM=BVPDSINI
//*-----------------------
```

```
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&DINIMB,DISP=(OLD,PASS)
//PACDBB   DD DSN=&INDUN..&BAS.0BB(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPABB,
//         DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDRU   DD SYSOUT=&OUT
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 14. DXBJ - Journal extraction for update

DXBJ - Introduction

The DXBJ procedure extracts, from the DSMS journal file, all the transactions corresponding to a date/time interval or to a given user, and transforms them into update transactions.

Execution conditions

None.

Abnormal execution

Refer to Chapter 'The batch procedures', Subchapter 'Abnormal execution'.

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

DXBJ - Input / Processing / Results

USER INPUT

One '*'-line is required:

Pos.	Len.	Value	Meaning
2	1	'*'	line code
3	8	uuuuuuuu	DSMS User code
11	8	pppppppp	User password
			OPTIONAL:
19	3	ppp	Product code
22	2	su	Subsidiary code
24	1	'F' or 'E'	Language code
		'	USERS/PASSWORDS IN OUTPUT TRANSAC.

One line per extraction request:

Pos.	Len.	Value	Meaning
2	1	'K'	Line code

Pos.	Len.	Value	Meaning
3	1	' '	List of selected transactions
		'N'	No list
4	8	CCYYMMDD	Starting date for selection
12	8	CCYYMMDD	Ending date for selection
20	6	HHMMSS	Starting time for selection
26	6	HHMMSS	Ending time for selection
32	8	uuuuuuuu	Selected user code
40	1	' '	User codes present in journal file without password.
		'T'	User codes present in journal file with passwords if sufficient authorization.
		'1'	User code and password, detailed in next columns.
41	8	uuuuuuuu	User code for output transactions (if column 40 = 1)
48	8	mmmmmmmm	Password for output transactions (if column 40 = 1)

REPORT

Extraction report and, upon request, the list of formatted transactions.

RESULT

A DSMS update transactions file to be used as input to the DUPT procedure. An 'N' is entered in column 36 of the user line for DUPT not to sort these transactions.

DXBJ - Description of steps

Extraction and formatting of transactions: PDS700

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDBJ	&INDUN..&base.BJ	Input	Archived DSMS journal
PACDMB	&&DXBJMB	Input	User transactions
PACDIM	&PACDIM	Output	Update transaction file for DUPT
PACDRK		Report	Extraction review
PACDSK		Report	Transaction printout

Return codes:

- 0: No error
- 8: Error on the user '*' line or parameter line.

  The environment definition is missing.
- 12: File access error.

  The technical record is missing.

DXBJ - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - EXTRACTION OF DSMS JOURNAL -                        *
//*********************************************************************
//BVPDXBJ  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//  SPAIM='(TRK,(15,5))',   SPACE OF EXTRACTED TRANSACTIONS
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK'             WORK UNIT
//*****************************************************************
//*   FORMAT OF TRANSACTIONS AT INPUT :
//*   .. A DSMS USER AND PASSWORD LINE
//*  COL 02     : *
//*  COL 03     : DSMS USER CODE
//*  COL 11     : PASSWORD
//*  COL 19     : PRODUCT CODE       (OPTIONAL)
//*  COL 22     : SUBSIDIARY CODE    (OPTIONAL)
//*  COL 24     : LANGUAGE           (OPTIONAL)
//*   .. COMMAND LINE(S) FOR EXTRACTION
//*  COL 02     : K
//*  COL 03     : ' ' SELECTED TRANSACTIONS LIST
//*             : 'N' NO LIST OF SELECTED TRANSACTIONS
//*  COL 04-11  : STARTING DATE (CCYYMMDD)
//*  COL 12-19  : ENDING   DATE (CCYYMMDD)
//*  COL 20-25  : STARTING HOUR (HHMMSS)
//*  COL 26-31  : ENDING   HOUR (HHMMSS)
//*  COL 32-39  : USER CODE
//*                                                                   *
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&DXBJMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
```

```
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDS700 EXEC PGM=BVPDS700
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDMB   DD DSN=&&DXBJMB,DISP=(OLD,PASS)
//PACDIM   DD DSN=&&PACDIM,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAIM,
//         DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PACDBJ   DD DSN=&INDUN..&BAS.0BJ(0),DISP=SHR
//PACDRK   DD SYSOUT=&OUT
//PACDSK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 15. DREN - Code and keyword update

DREN - Introduction

The Code and Keyword Update procedure (DREN) is used to define new codes (table or site) or new keywords to replace those defined and used until then in the tables, thesaurus, and entities.

Execution condition

This procedure works from a sequential backup and/or an archived journal, and must therefore be preceded by a backup and/or an archiving.

Abnormal execution

See Subchapter 'Abnormal Execution', in Chapter 'The batch procedures'.

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

DREN - Input / Processing / Results

USER INPUT

One '*' line (required):

Col.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	DSMS User Code
11	8	pppppppp	Password
			OPTIONAL
19	3	ppp	Changes made on the entities which depend on the product code 'ppp'
		'***'	Changes made on the entities which depend on all the product codes
22	2	ss	Changes made on the entities which depend on the subsidiary code 'ss'
		'**'	Changes made on the entities which depend on all the subsidiary codes
24	1	'E' or 'F'	Language code

Col.	Len.	Value	Meaning
			REQUIRED: AT LEAST ONE OF THESE AREAS SET TO '1'
25	1	' '	No change concerning the backup
		'1'	Changes concerning the backup
26	1	' '	No change concerning the archiving
		'1'	Changes concerning the archiving

Command lines (500 maxi)

Col.	Len.	Value	Meaning
2	3	'Txx'	table choice (idem TP)
		'Kxx'	keyword choice (with xx = 'T ' for technical keywords, xx = 'E ' for native keywords and xx = 'Cl' for keywords of change l language)
		'S '	site choice
5	13		old code
18	13		new code

Notes:

- The codes (old and new) must be preceded by 'C', 'E' or 'S' for the TST table, by 'C' or 'E' for the TGR and TTY tables, and by 'F' or 'R' for the TAT table.
- It is not possible to invert two codes (for example, change 'AA' to 'BB', and 'BB' to 'AA'). However, it is possible to rename a code (with an unknown one), and to reuse the old code to transform other codes (for example: 'AA' becomes 'BB' while 'CC' and 'DD' become 'AA'; in this case the command AA/BB must be written before CC/AA and DD/AA).
- The new codes assigned to products, subsidiaries or sites must not already exist (in the same subsidiary for a site).
- The two parts of the site code (9 and 3 characters) cannot be modified separately.
- For the TVE table, you can request the following updates:
  - Technical release alone
  - Technical release and release
  - Technical release, release and hardware
  - Technical release, release, hardware and version (with or without language code)
  - Release alone

- Hardware alone
- Version number (with or without language code)

Isolated parts should be aligned as if the other parts were there.

Ascending consistency checks are performed. The changes requested on the preceding lines must be taken into account.

- The label associated with the new code can be either that of the old code or that of the new code if it already existed. This choice is made while the file is sorted and is therefore unpredictable.
- For tables depending on a product (TOP, TPH and TVE), the product's code must be clearly specified on the '*' line.

PRINTED OUTPUT

Report on changes concerning the backup and/or the archiving.

Note on counters:

They count the total number of updates but not the number of modified records (there can be several modifications on the same record).

RESULT

If the change was made on the archive (1 in column 26), a new version of the Journal's sequential backup is produced.

If the change was made on the Database backup (1 in column 25), the result is a new version of the Database sequential backup which should be reorganized via the DREO procedure before being restored.

Return code

Code	**Meaning**
0	OK
8	Error on the '*' line or on a command line
10	Absence or invalid value for backup 'top'
11	Absence or invalid value for archive 'top'
12	Input/Output error or inconsistent DSMS base
	Invalid absence of backup/archive 'tops'
16	Sort error

DREN - Description of steps

This procedure calls a single program (PDSMS) which is used as a flow monitor for various programs considered as sub-routines of this monitor. It includes the following steps:

Updates: PDSMS

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDC	&INDUV..&base.DC	Input	VA Pac element file
PACDDE	&INDSV..BVPDE	Input	Error messages
PACDDX	&INDUV..&base.DX	Input	Cross-references
PACDBB	&INDUN..&base.BB(0)	Input	DSMS backup
PACDBJ	&INDUN..&base.BJ(0)	Input	DSMS archiving
PACDMB	&&PACDMB	Input	User queries
PACDW0		Workfile	Update requests
PACDW1	&&W1	Workfile	Partial backup (sorted)
PACDW2	&&W2	Workfile	Partial backup (not sorted)
PACDB3	&INDUN..&base.0BB(+1)	Output	Modified backup
PACDJB	&INDUN..&base.0BJ(+1)	Output	Modified archive
PACDIA		Report	Branching report
PACDIK		Report	List of commands on the backup
PACDJK		Report	Update report (backup)
PACDIS		Report	Merging report (backup)
PACDKK		Report	List of commands on archiving
PACDLK		Report	Update report (archive)
SORTWK01 SORTWK02 SORTWK03		Sort	

DREN - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - CHANGE OF TABLE AND SITE CODES, AND KEYWORDS -      *
//*********************************************************************
//BVPDREN  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
```

```
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//  SPAWK='(TRK,(60,5),RLSE)',  WORK SPACE (DA + DC)
//  UWK=$UWK,              UNIT OF WORK FILES
//  LNG=$LNG,    LANGUAGE OF MONITOR (E FOR ENGLISH F FOR FRENCH)
//  CYL=3,                 SORTWORK SPACE
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  DSCB='$DSCB',          DSCB MODEL FILE
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(10,2),RLSE)',  SPACE OF BACKUP (IF DISK)
//  SPABJ='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//*****************************************************************
//* INPUT :
//* .. IDENTIFICATION LINE
//*  COL 02     : *
//*  COL 03     : DSMS USER CODE
//*  COL 11     : PASSWORD
//*  COL 19-21  : PRODUCT CODE OR '***'
//*  COL 22-23  : SUBSIDIARY CODE OR '**'
//*  COL 24     : LANGUAGE CODE
//*  COL 25     : MODIFICATIONS ON SAVE (1, )
//*  COL 26     : MODIFICATIONS ON ARCHIVE (1, )
//* ..  MODIFICATION(S) COMMAND LINE(S)
//*  COL 02-04  : TYPE OF MODIFICATION
//*  COL 05-17  : OLD CODE
//*  COL 18-30  : NEW CODE
//*
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//VERIFY EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDSMS  EXEC PGM=BVPDSMS&LNG,REGION=4096K
//*--------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
```

```
//PACDMB   DD DSN=&&PACDMB,DISP=SHR
//PACDBB   DD DSN=&INDUN..&BAS.0BB(0),DISP=OLD
//PACDBJ   DD DSN=&INDUN..&BAS.0BJ(0),DISP=OLD
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDW0   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6160
//PACDW1   DD DSN=&&W1,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAWK,
//         DCB=(RECFM=VB,BLKSIZE=6022,LRECL=354)
//PACDW2   DD DSN=&&W2,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAWK,
//         DCB=(RECFM=VB,BLKSIZE=6022,LRECL=354)
//PACDB3   DD DSN=&INDUN..&BAS.0BB(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//         DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDJB   DD DSN=&INDUN..&BAS.0BJ(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPABJ,
//         DCB=(&DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//PACDIA   DD SYSOUT=&OUT
//PACDIK   DD SYSOUT=&OUT
//PACDJK   DD SYSOUT=&OUT
//PACDKK   DD SYSOUT=&OUT
//PACDLK   DD SYSOUT=&OUT
//PACDIS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//DEL10  EXEC PGM=IEFBR14,COND=(10,NE,PDSMS)
//*---------------------------------------
//DDBB     DD DSN=&INDUN..&BAS.0BB(+1),DISP=(OLD,DELETE)
//DEL11  EXEC PGM=IEFBR14,COND=(11,NE,PDSMS)
//*---------------------------------------
//DDBJ     DD DSN=&INDUN..&BAS.0BJ(+1),DISP=(OLD,DELETE)
//DEL12  EXEC PGM=IEFBR14,COND=(12,NE,PDSMS)
//*---------------------------------------
//DDBB     DD DSN=&INDUN..&BAS.0BB(+1),DISP=(OLD,DELETE)
//DDBJ     DD DSN=&INDUN..&BAS.0BJ(+1),DISP=(OLD,DELETE)
```

Chapter 16. DPDF - Pre-processing of generated DAF programs

DPDF - Introduction

The DPDF procedure processes user generated programs that contain SQL requests for Database access through DAF operators.

Execution condition

None.

Implementation

The DPDF procedure may be executed in several ways:

- Either after a program generation via GPRT, whose generated output is used as input to DPDF, before being passed on for compilation or storing in a source-program library.
- Or by a call to the procedure in the optional before/after control cards of the program. In this case, the correct JCL must have been entered in the selected options, which are updated in Administrator Workench, in the 'Optional Command Lines Sets' tab.

DPDF - Input / Processing / Results

USER INPUT

It is the COBOL source of the programs containing DAF operators which must be solved by the pre-processor before being compiled.

Each program contains, after the IDENTIFICATION DIVISION line, a command line for the pre-processor:

Pos.	Len.	Value	Meaning
1	6	nnnnnn	COBOL line number
7	1	'*'	Comments
8	5	'TP '	On-line program, or
		'BATCH'	Batch program
13	6	'LIB:'	Fixed label
19	3	bbb	Library code

Pos.	Len.	Value	Meaning
22	1	blank	Not used
23	5	nnnns	Session number - Session status
28	1	blank	Not used
29	2	--	Generation variant(s)
31	5	'AR:'	Fixed label
36	1	l	Database language code
37	5	'SC:'	Batch language program skeleton
		'SG:'	OLSD program skeleton
		'SR:'	COBOL Generator program skeleton
42	1	l	Skeleton language
43	1	blank	Not used
44	6	'SINGLE'	Single quotes, or
		'DOUBLE'	Double quotes

Examples:

```
000020*TP    LIB: APP 2345  00 AR: F SG: F SINGLE
000020*BATCH LIB: APP 2300T 4  AR: F SC: F DOUBLE
```

This line is automatically generated by the GPRT procedure.

PRINTED OUTPUT

The procedure prints the list of errors, if any.

RESULT

The result of the execution is a COBOL source file in which all DAF operators have been solved, and all the calls to Database batch or on-line access routines have been generated.

DPDF - Description of steps

The DPDF procedure calls a single program which acts as a flow monitor for various programs, considered as sub-programs of this monitor. It includes the following step:

Generated program's pre-processor: DAFD10

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
DAF80	&&DPDFMB	Input	Generated programs
COB80	&&DAFGEN	Output	Generated programs to be compiled
DAFREP		Report	Execution report

Note: If the generated stream contains the compilation's control cards (when DPDF is executed after GPRT), the DSN (&&DAFGEN) may be replaced with the sending of SYSOUT=(&OUT,INTRDR) to the MVS machine's Internal Reader.

DPDF - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*              - ACCESS FACILITY PRE-PROCESSING -                   *
//*********************************************************************
//BVPDPDF  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//         INDSV='$INDSV',      INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',      INDEX OF SYSTEM NON VSAM FILES
//         INDUV='$INDUV',      INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',       USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',       SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//         OUT=$OUT,              OUTPUT CLASS
//         SPAMB='(TRK,(150,15))', SPACE OF GENERATED PROGRAMS
//         UWK=$UWK               WORK UNIT
//*****************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&DPDFMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//DAFD10 EXEC PGM=BVPDAFP
```

```
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//DAF80    DD DSN=&&DPDFMB,DISP=(OLD,DELETE)
//COB80    DD DSN=&&DAFGEN,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//DAFREP   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 17. DUPD - Batch update from DAF tables

DUPD - Introduction

The DUPD procedure performs the batch update of the DSMS Database from a sequential file mirroring the DAF tables.

Its operating principle is quite similar to that of the DUPT procedure, except for the format of the input transactions.

Execution condition

Refer to the chapter dedicated to DUPT.

Abnormal execution

Refer to the chapter dedicated to DUPT.

DUPD - Input / Processing / Results

USER INPUT

The sequential file of input transactions is produced by a DAF extractor program. Its records mirror the DAF tables (described in the DAF TABLES Manual).

Pos.	Length	Meaning
1	1	Transaction code (C, M, X, D or A, B)
2	10	DAF table code
12	299	DAF table contents (described in the DAF tables Manual).

UPDATE RULES

Update transactions are not sorted.

Each set of transactions impacting a library or session must be preceded by an ASSIGN table code line.

Pos.	Len.	Value	Meaning
2	10	'ASSIGN'	Table code
12	8	uuuuuuuu	User code

Pos.	Len.	Value	Meaning
20	8	pppppppp	Password
28	3	ppp	Product code
31	2	ss	Subsidiary code

PRINTED OUTPUT

Refer to the description of the DUPT output.

RESULT

Refer to the description of the DUPT result.

DUPD - Description of steps

Database integrity check: PDSBAS

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input	Data file
PACDDE	&INDSV..BVPDE	Input	Error message file
PACDRS		Report	Validity report

Return code

This utility sends a return code 4 and causes an ABEND if the Database is invalid.

Transaction formatting: PDS900

Code	Physical name	Type	Label
PACDGY	&DAFINPUT Length=382	Input	Update transactions
PACDIM	&&PACDIM Length=250	Output	Formatted transactions

Update of the DSMS Database: PDSUP0

Code	Physical name	Type	Label
PACDDA	&INDUV..&base.DA	Input Output	Data file

Code	Physical name	Type	Label
PACDDC	&INDUV..&base.DC	Input Output	VisualAge Pacbase elements file
PACDDX	&INDUV..&base.DX	Input Output	Cross-references file
PACDDE	&INDSV..BVPDE	Input	Error messages file
PACDIM	&&PACDIM	Input	Update transactions output by DEXP
PACDDJ	&INDUV..&base.DJ	Input	Journal
PACDRP		Report	Update report
SORTWK01 SORTWK02 SORTWK03		Tri	

Return codes:

- 0 : No error on files
- 8 : Error on user identification line or parameter
- 12 : Input/output error on a file

DUPD - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - BATCH UPDATE FROM DAF TABLES -                      *
//*********************************************************************
//BVPDUPT  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  DAFINPUT=,             INPUT TRANSACTIONS DSN
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//  UWK='$UWK',            SORTWORK SPACE
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  SPAIM='(TRK,(100,10),RLSE)'   TRANSACTION SPACE
//*****************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
```

```
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDX),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFDE),DISP=SHR
//PDSBAS EXEC PGM=BVPDSBAS
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDS900 EXEC PGM=BVPDS900,COND=(0,NE,PDSBAS),REGION=0K
//*-------------------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACDGY   DD DSN=&DAFINPUT,DISP=SHR
//PACDIM   DD DSN=&&PACDIM,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAIM,
//            DCB=(RECFM=FB,LRECL=250,BLKSIZE=3500)
//PDSUP0 EXEC PGM=BVPDSUP0,REGION=4096K,COND=(0,NE,PDSBAS)
//*-----------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDIM   DD DSN=&&PACDIM,DISP=(OLD,DELETE)
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&BAS.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//PACDRP   DD SYSOUT=&OUT
//SYSIN    DD DUMMY,DCB=BLKSIZE=80
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 18. Installation

Parameterization

SMP/E context

VA Pacbase, DSMS and Pactables use common batch and online load modules.

These load-modules are supplied in the HBVP350 root FMID.

The components specific to VA Pacbase are supplied in the JBVP351 dependent FMID.

The components specific to DSMS are supplied in the JBVP352 dependent FMID.

The components specific to Pactables are supplied in the JBVP353 dependent FMID.

To install a dependent FMID, the HBVP350 root FMID must be simultaneously or previously installed.

If the HBVP350 root FMID is already installed, only the dependent JBVP35x FMID must be installed.

On the installation cartridge, the HBVP350 root FMID and the JBVP35x dependent FMID are systematically provided.

The RECEIVE, APPLY and ACCEPT JCLs, that are supplied, are to be used for a simultaneous installation of the two FMIDs. They must be adapted if the HBVP350 root FMID is already installed.

Installation preparation

The preparation to the installation process consists of three phases:

- Receiving the cartridge with the SMP/E (System Modification Program/Extended) utility in some dedicated PDS.

  This phase must be performed by the system staff who are accustomed to installing IBM products with SMP/E.
- Allocating the PDS file where all the installation and operation JCLs will be saved.
- Executing the installation JCL from the PDS members created during the first phase.

Remark for a previous SMP/E installation

The following SMP/E phase implies that the SMP/E context is empty for the product: either the product has never been installed with SMP/E, or the files related to SMP/E have been reinitialized for the previous version, or you want to create a new SMP/E environment for this version.

Otherwise, if you want to re-use the SMP/E files of the previous version, you can execute, before phase 1, the SMP/E utility described in the Appendix at the end of this manual.

First phase

You will find details in the document 'Program Directory for DSMS' specific to SMP/E.

When the installation takes place in a virgin environment, the necessary SMP/E environment must be implemented and the HBVP350 and JBVP352 FMIDs must be installed with the supplied JCLs.

This stage consists of the following steps:

- Downloading, from the cartridge, the sample JCLs necessary to the execution of steps 2, 3, and 4. To do so, follow the instructions specified in the 'Program Directory for DSMS' document.
- Setting up the SMP/E environment and the 'Target zone' and 'Distribution zone' files of the HBVP350 root FMID:
  - Defining the SMP/E cluster libraries (BVP1DCSI)
  - Initializing the SMP/E libraries (BVP2ICSI)
  - Allocating the SMP/E work files (BVP3ALLO)
  - Assigning the FMID (BVP4DEFZ)
  - Creating the DDDEF input of technical files (BVP5DDEF)
  - Defining the 'Target zone' and 'Distribution zone' files (BVP6DDEF)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP7ALLO)
- Implementing the 'Target zone' and 'Distribution zone' files of the JBVP352 dependent FMID:
  - Defining the 'Target zone' and 'Distribution zone' files (BVP8DDD)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP9DAL)
- Installing the components of the HBVP350 and JBVP352 FMIDs:
  - Execution of RECEIVE (BVPDREC)
  - Execution of APPLY (BVPDAPP)
  - Execution of ACCEPT (BVPDACC)

When the installation is executed in an environment where the HBVP350 FMID and a JBVP35x FMID are already installed, you just have to install the dependent FMID JBVP352 of DSMS.

Then execute the steps 3 and 4 once the necessary JCLs are downloaded as described in the step 1 and modify the RECEIVE, APPLY and ACCEPT JCLs to remove the HBVP350 FMID from the concerned lines.

Once this phase has been executed, all the components required for the installation are to be found in the following PDSs ; 'hlq' indicates the common prefix of the elements supplied (High-Level Qualifier):

- hlq.SBVPINST: all the 80-long files, including:
  - The initial installation JCL (BVPDINIT)
  - A file which contains the installation and operation JCLs and procedures (BVPDTTAL)
  - The sample SMP/E JCLs
- hlq.SBVPDF2: BB initialization files
- hlq.SBVPMBR8: batch load-modules
- hlq.SBVPMTR8: on-line load-modules
- hlq.SBVPDF5: error messages file

Second phase

This stage is optional but recommended. It consists in allocating a PDS file with the following characteristics:

- Lrecl=80
- Size: around 100 tracks of a 3390 disk, 30 pads directory.

Third phase

The third phase involves copying the 'hlq.SBVPINST(BVPDINIT)' JCL into the PDS mentioned in the second phase, modifying its parameters to match the constraints of the site and executing it so as to obtain a complete installation and operation JCL.

The BVPDINIT JCL executes the BVPMMJCL program loaded in the hlq.SBVPMBR8 PDS.

It must be completed as follows:

- Fill in '&hlq' with the value of the 'hlq' parameter used in the SMP/E first phase.
- In the '//SYSUT2 DD DSN=' field, enter the name of the file in which the complete JCL is to be saved.

This file can be either a PDS member initially created to save all the JCLs, or a sequential file selected by the user.

- Enter the parameters (see details in the next chapter).

The BVPMMJCL program execution must be saved: it can be used for a further re-installation.

Initial JCL

```
//VAPACDSM  JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//MM1JCL EXEC PGM=BVPMMJCL
//STEPLIB  DD DISP=SHR,DSN=&HLQ.SBVPMBR8
//         DD DISP=SHR,DSN=---.---.---    LE LIBRARY
//SYSOUT   DD SYSOUT=A
//SYSUT1   DD DSN=&HLQ.SBVPINST(BVPDTTAL),DISP=SHR
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT8   DD DUMMY,DCB=BLKSIZE=1370
//SYSUT9   DD DUMMY,DCB=BLKSIZE=1370
//*********************************************************************
//*                                                                   *
//*     CREATION OF INSTALLATION JCL THROUGH 'BVPMMJCL'               *
//*     ----------------------------------------------                *
//*                                                                   *
//*     MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ASKING,           *
//*     IF NECESSARY, A SELECTION OF PARTS OF INSTALLATION JCL        *
//*     (JCL MODULES), BY GIVING THE APPROPRIATE VALUES TO THE        *
//*     INSTALLATION PARAMETERS, AND, IF NECESSARY, BY SPECIFYING     *
//*     THE LINES TO BE ADDED AT THE BEGINNING OR AT THE END OF       *
//*     EACH JCL MODULE.                                              *
//*********************************************************************
//SYSPRM DD DUMMY
//SYSUT2   DD ---------   PDS MEMBER OR SEQUENTIAL FILE RECEIVING
//*                       THE INSTALLATION JCL (LRECL=80)
//SYSIN DD *
===PRM PRFJ=PAC           .JOB NAMES PREFIXES (3 CHARACTERS)
===PRM CCPT=<>            .JOB ACCOUNTING CODES (JOB CARDS)
===PRM CLASSJ=1           .JOB EXECUTION CLASS (JOB CARDS)
===PRM MSGCL=A            .JCL OUTPUT CLASS (MSGCLASS)
===PRM OUT=A              .JOB SYSOUT CLASS
===PRM INDSV='EXP.DSM35'  .DSMS SYSTEM VSAM FILES INDEXES
===PRM INDUV='CICS.DSM35' .DSMS USER VSAM FILES INDEXES
===PRM INDSN='EXP.DSM35'  .DSMS SYST.NON VSAM FILES INDEXES
===PRM INDUN='PDS.R35'    .DSMS USER NON VSAM FILES INDEXES
===PRM VOLSV=<>           .VOL.OF DSMS SYSTEM VSAM FILES
===PRM VOLSN=<>           .VOL.OF DSMS SYSTEM NON VSAM FILES
===PRM VOLUV=<>           .VOL.OF DSMS USER VSAM FILES
===PRM VOLUN=<>           .VOL.OF DSMS USER NON VSAM FILES
===PRM SCAT=<>            .CATALOG OF DSMS SYSTEM VSAM FILES
===PRM VCAT=<>            .CATALOG OF DSMS USER VSAM FILES
===PRM UWK=SYSDA          .WORK UNIT
===PRM UNITSN=SYSDA       .NON VSAM DSMS SYSTEM FILES UNIT
===PRM UNITUN=SYSDA       .NON VSAM DSMS USER FILES UNIT
===PRM LNG=E              .MONITOR LANGAGE (E=ENGLISH, F=FRENCH)
```

```
===PRM HLQ='HLQ'          .HIGH LEVEL QUALIFIER OF LOAD MODULES
===PRM BAS='BAS'          .NAME OF DATABASE DEVELOPMENT (3 C)
===PRM BIBP='SYS1.PROCLIB' .PROCEDURE LIBRARY
===PRM BIBT='SYS1.SORTLIB' .SORT LIBRARY
===PRM DSCB='PDS.DSCB'     .DSCB MODEL FILE DSNAME
===PRM CSDL='CICS330.LOADLIB'  .DFHCSDUP STEPLIB DSN
===PRM DFHCSD='PAC.DFHCSD'     .DSMS CSD DSN
===PRM LIST=<>                 ."LIST" NAME
===PRM GROUP='PACDSMS'         ."GROUP" NAME FOR PROGRAMS
===PRM GROUT='PACDSMS'         ."GROUP" NAME FOR TRANSACTION
===PRM GROUF='PACDSMS'         ."GROUP" NAME FOR FILES
===BEGMOD
./ ADD NAME=$ZMODUL
/*
//
```

Installation of complete JCL

The BVPMMJCL module reads the JCL skeleton file and produces a complete JCL. It allows you to:

- Select portions of the JCL skeleton, which are called 'JCL modules',
- Parameterize the skeleton in order to obtain a JCL which requires a minimum of modifications to be operational,
- Add lines before and/or after the JCL modules to separate these modules.

This step can be executed as many times as necessary to generate a complete JCL.

USER INPUT

Command	Parameters	Comments
===PRM	PPPP=pppp (1)	Parameter
===SELM	jcl1 jcl2	Selected modules
===BEGMOD		Insertion of lines at the beginning of module
....1		Lines to be added before each module
....n		
===ENDMOD		Insertion of lines at the end of module
....1		Lines to be added after each module
....n		

(1) PPPP = parameter name

pppp = parameter value

Notes

- Lines ===PRM

  You can add a comment but it must be preceded by a dot and it should not exceed the column 72.

  The default values are only examples. You must therefore enter values which correspond to the choices on your site.
- Lines ===SELM

  These lines can be used to select modules.

  As the standard installation provides all the modules, no particular module is selected.
- Lines ===BEGMOD

  ./ ADD NAME=$MODULE

  As a result, a line is inserted before each JCL module, in the form:

  ./ ADD NAME=<JCL-module>

RESULT: complete JCL

The file obtained in SYSUT2 contains all the installation and operation JCLs. This file must be processed via an editor to enable the installation process.

Two operations must be performed on the complete JCL:

1. Global modifications (if necessary)

Adaptations can be performed on all the JCLs.

VSAM catalogues appear as comments in the JCL obtained after the installation:

- In the DELETE/DEFINE*/
- In the JCL STEPCATs
- In the procedures' parameters

When these parameters are not required on the site, the resulting JCL can remain as it is.

When these parameters are required on the site, the affected lines should be changed into command lines. To do so, you must:

- Transform all '//*:' into '//',
- And then replace '/*:' and '*/' with blanks.

Caution: SMS

- In the installation jobs which include GenerationDataGroup allocation, you must delete the lines DD //GDGMOD from the definition IDCAMS.

- If the UNIT and VOL parameters cannot be used on the site, you can delete them in the whole JCL via an exclude command (EXCLUDE command in TSO/EDIT).

In most cases, it is recommended to perform general modifications on JCLs before the JCL splitting operation.

Caution: LSR

JCLs are standardly provided with the LSR option for the batch optimization of VSAM files access.

If the LSR option is not implemented on site, you must modify the following lines in the procedures:

```
//xxLSR DD DSN=&INDxx..file,DISP=SHR
//PACxx DD SUBSYS=(&LSR,'DDNAME=xxLSR','BUFND=10','BUFNI=10
```

by a line:

```
//PACxx DD DSN=&INDxx..file,DISP=SHR.
```

2. JCL splitting

Before each module of a standard complete JCL, there is a ./ ADD NAME=<JCL-module> line, where <JCL-module> is the code of the ===MOD line that is found (see the following table of JCL modules).

So the complete JCL can be split in as many members as JCL modules in a PDS. The complete JCL file is to be used as SYSIN for the PDS update utility: IEBUPDTE.

Note: Because of this default option, all ’./’ characters found in JCL modules containing IEBUPDTE were replaced with ’:/’.

Once the JCL is split, the replacement must be done the other way round before executing jobs which contain IEBUPDTE.

REPORT

BVPMMJCL outputs a list for each JCL module created, with the parameters taken into account and according to the variants requested.

Note:

Since the JCL skeleton parameters are formatted as $xxxx, if BVPMMJCL encounters, upon execution, a $ character which does not correspond to a

defined parameter, it sends error messages such as: 'Unknown symbolic parameter' or 'Invalid position or length' or 'Syntax error in symbolic parameter'.

These messages do not stop the execution and should be ignored: they apply to '$' characters present in the flow processed by BVPMMJCL but which are NOT parameters.

CODING OF MM1JCL COMMANDS

```
===SELM mmmm1 mmmm2 ...   .Selection of JCL modules
                           mmmm1 = name of JCL module
                           mmmm2 = name of JCL module
                           etc.
                           If there is no ===SELM line, all
                           the JCL modules are selected.
===PRM  PPPP=pppp         .Parameter
                           PPPP = parameter name
                           pppp = parameter value
```

CAUTION

On ===PRM or ===SELV lines, comments may be entered. They should be preceded by a period, and not exceed column 72.

```
===BEGMOD         Insertion of lines at beginning of module.
....1         )
.....         ) Lines to be inserted before each module
....n         )
===ENDMOD         Insertion of lines at end of module.
....1         )
.....         ) Lines to be inserted after each module
....n         )
```

Installation default options

- Parameters (===PRM):

  The values indicated are examples; they should be replaced according to the site's specific needs.
- Modules (===SELM):

  No selection; all modules (corresponding to the variants) are copied.
- JCL module first line (===BEGMOD):

  A line: ./ ADD NAME=$MODULE

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

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

JCL modules

```
===SELM mmmm1 mmmm2 ... mmmmn
```

mmmm	Contents	Nature
	CICS UPDATE:	
DCICSD	CSD update	JCL
	DSMS INSTALLATION:	
DI1SY	Allocation/loading of parameters PDS	-
DI2PRE	GDG allocation/backup file initialization	-
DI4DE	DE file installation	-
DI5IPRO	Allocation of operations procedures	-
DI5PRO	Cataloging of operations procedures	-
DI6DS	Restoration of test deck (DRST)	-
	SAMPLE JOBS FOR TESTS	
JCLDARC	Journal archiving	JCL
JCLDEXT	Extraction (DEXT) and possible update (DUPT)	-
JCLDRST	Database Restoration	-
JCLDINS	List of program dates	-
JCLDSAV	Database Backup	-
JCLDXBJ	Journal extraction in the form of update transactions	-
JCLDAF	Execution of a batch program for DSMS access via DAF	-

JCL parameterization

Code	Meaning	Default
	ON THE JOB CARDS	
PRFJ	3-character job name prefix	PAC
CCPT	Job accounting code	<>
CLASSJ	Job execution class	1
MSGCL	JCL output class	A
	CODING OF FILE DSNs	
	All DSMS files (except load-module libraries) are named as follows:	
	INDUV.$base.ff : User VSAM file	
	INDUN.$base.ff : User non-VSAM file	
	INDSV.BVPDE : Error message VSAM file	
	INDSV.$.base.DF : DAF VSAM	

Code	Meaning	Default
	INDSN.BVPSY : non-VSAM SY PDS file	
	IND-- Index of file names:	
INDSV	VSAM system	EXP.PAC80
INDSN	Non-VSAM system (SAM, PDS)	EXP.PAC80
INDUV	VSAM user	CICS.PAC80
INDUN	Non-VSAM user (SAM)	PAC.PAC80
LNG	Langage	E
HLQ	Common prefix of the elements supplied with SMP/E (30 characters maximum)	HLQ
BAS	Database code (3 characters)	BAS
	ON THE DD CARDS	
OUT	SYSOUT class	A
UWK	UNIT of work files used	SYSDA
UNITSN	UNIT of non-VSAM system files	SYSDA
UNITUN	UNIT of non-VSAM user files	SYSDA
VOLSN	Volume name of non-VSAM system files	<>
VOLSV	Volume name of VSAM system files	<>
VOLUN	Volume name of non-VSAM user files	<>
VOLUV	Volume name of VSAM user files	<>
	CICS CSD UPDATE PARAMETERS (CICS/ESA V3 RDO only)	
CSDL	DFHCSDUP STEPLIB DSN	'CICS311.LOADLIB'
DFHCSD	VA Pac CSD DSN for update	'PAC.DFHCSD'
GROUP	Program CSD input group	PACDSMS
GROUT	Transaction CSD input group	PACDSMS
GROUF	File CSD input group	PACDSMS
LIST	List to which the DSMS group is to be added	<>
	OTHER PARAMETERS	
VCAT	DSNAME of User Files VSAM Catalog	
SCAT	DSNAME of System Files VSAM Catalog	
BIBP	DSNAME of the procedure library in which the DSMS procedures are to be catalogued.	SYS1.PROCLIB
BIBT	DSNAME of the sort library.	SYS1.SORTLIB
DSCB	DSNAME of the DSCB MODEL file.	PDS.DSCB

Note: The '<>' indicates a required parameter.

VERY IMPORTANT:

- The parameter values which contain special characters must be bounded by quotes.
- Comments on ===PRM lines should not exceed column 72. They should be preceded by a '.'.

JCL module separators

Lines before and after JCL modules

```
===BEGMOD
....1    )
.....    ) Lines to be inserted before each JCL module
....n    )
===ENDMOD
....1    )
.....    ) Lines to be inserted after each JCL module
....n    )
```

Lines may be inserted as input to BVPMMJCL if the default option is not appropriate (see Subchapter 'Installation default options' above).

The purpose of these lines is to execute the separation of the JCL file created by the BVPMMLCL utility into as many members as there are JCL modules.

This utility adds ....1 to ....n lines in front of each JCL module and ....1 to ....n lines to the end of each JCL module.

Installation steps

Once all JCL is obtained, the installation of the DSMS system is broken down into nine phases:

1. Update of CICS,

2. Allocation/Loading of the SY parameters PDS,

3. GDG allocation/Initialization of backup files,

4. Definition/Loading of the error message file/HELP function,

5. Cataloging of operations procedures,

6. Loading of the DSMS test deck files,

7. CICS modification,

8. Additional user operations.

9. Additional Security System Interface

CICS CSD update

DCICSD module: '$prfj.0CI' job

- Transaction:

  1 user transaction code: $base$lng

- Programs:

  named BVP00nn

  (See complete list in Chapter 'DSMS components' Subchapter 'ON-LINE PROGRAM LIBRARY').

- Files:
  - 2 DSMS files:

    BVPDE : Error messages/HELP function,

    $base.DF : Work file for DSMS Access Facility (DAF)
  - 4 user files :

    $base.DA : Data,

    $base.DX : Cross-references,

    $base.DC : VisualAge Pacbase elements,

    $base.DJ : Update Journal.

Notes on installation:

Entries are updated in the CSD by the $prfj.0CI job of the ===MOD DCICSD JCL module.

The option specifying the Dynamic Backout is compulsory for the DA, DC, DX and DJ files.

The number of STRINGs, INDEX BUFFER and DATA BUFFER is the same as the minimum number necessary for the system.

The CICS parameter 'SPOOL=YES' is compulsory to submit batch jobs on-line (choice : LVQ).

Execution JCL

```
//$PRFJ.0CI JOB ($CCPT),'PACDSMS DFHCSDUP',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//*********************************************************************
//* DSMS                                                              *
```

```
//*                                                                   *
//*              - 'DFHCSD' BATCH UPDATE -                            *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
*********************************************************************
*   CSD CICS : PACBASE    DSMS                                      *
*********************************************************************
DEFINE FILE(BVPDE) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS HELP)
       DSNAME($INDSV..BVPDE)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(NO) BROWSE(YES) DELETE(NO) READ(YES) UPDATE(NO)
DEFINE FILE($BAS.0DF) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS ACCESS FACILITY)
       DSNAME($INDSV..$BAS.0DF)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($BAS.0DA) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS DATA)
       DSNAME($INDUV..$BAS.0DA)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($BAS.0DC) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS PACBASE ELEMENTS)
       DSNAME($INDUV..$BAS.0DC)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($BAS.0DX) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS XREF)
       DSNAME($INDUV..$BAS.0DX)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
```

```
DEFINE FILE($BAS.0DJ) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS JOURNAL)
       DSNAME($INDUV..$BAS.0DJ)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
*********************************************************************
*   CSD CICS : PACBASE    DSMS   PROGRAMS                           *
*********************************************************************
DEFINE PROGRAM(BVPCHOI) GROUP($GROUP)
DEFINE PROGRAM(BVPCUAM) GROUP($GROUP)
DEFINE PROGRAM(BVPCUEV) GROUP($GROUP)
DEFINE PROGRAM(BVPCUMQ) GROUP($GROUP)
DEFINE PROGRAM(BVPCURQ) GROUP($GROUP)
DEFINE PROGRAM(BVPCUSI) GROUP($GROUP)
DEFINE PROGRAM(BVPMONI) GROUP($GROUP)
DEFINE PROGRAM(BVPMOSO) GROUP($GROUP)
DEFINE PROGRAM(BVPMWEB) GROUP($GROUP)
DEFINE PROGRAM(BVPTPDF) GROUP($GROUP)
DEFINE PROGRAM(BVP00AA) GROUP($GROUP)
DEFINE PROGRAM(BVP00AB) GROUP($GROUP)
DEFINE PROGRAM(BVP00BA) GROUP($GROUP)
DEFINE PROGRAM(BVP00B1) GROUP($GROUP)
DEFINE PROGRAM(BVP00B2) GROUP($GROUP)
DEFINE PROGRAM(BVP00B3) GROUP($GROUP)
DEFINE PROGRAM(BVP00B4) GROUP($GROUP)
DEFINE PROGRAM(BVP00B5) GROUP($GROUP)
DEFINE PROGRAM(BVP00B6) GROUP($GROUP)
DEFINE PROGRAM(BVP00EA) GROUP($GROUP)
DEFINE PROGRAM(BVP00E1) GROUP($GROUP)
DEFINE PROGRAM(BVP00E2) GROUP($GROUP)
DEFINE PROGRAM(BVP00E3) GROUP($GROUP)
DEFINE PROGRAM(BVP00E4) GROUP($GROUP)
DEFINE PROGRAM(BVP00E5) GROUP($GROUP)
DEFINE PROGRAM(BVP00E6) GROUP($GROUP)
DEFINE PROGRAM(BVP00FA) GROUP($GROUP)
DEFINE PROGRAM(BVP00FB) GROUP($GROUP)
DEFINE PROGRAM(BVP00HE) GROUP($GROUP)
DEFINE PROGRAM(BVP00JO) GROUP($GROUP)
DEFINE PROGRAM(BVP00KA) GROUP($GROUP)
DEFINE PROGRAM(BVP00K1) GROUP($GROUP)
DEFINE PROGRAM(BVP00K2) GROUP($GROUP)
DEFINE PROGRAM(BVP00K3) GROUP($GROUP)
DEFINE PROGRAM(BVP00LE) GROUP($GROUP)
DEFINE PROGRAM(BVP00LS) GROUP($GROUP)
DEFINE PROGRAM(BVP00MA) GROUP($GROUP)
DEFINE PROGRAM(BVP00PA) GROUP($GROUP)
DEFINE PROGRAM(BVP00P1) GROUP($GROUP)
DEFINE PROGRAM(BVP00QA) GROUP($GROUP)
DEFINE PROGRAM(BVP00QB) GROUP($GROUP)
DEFINE PROGRAM(BVP00QC) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q1) GROUP($GROUP)
```

```
DEFINE PROGRAM(BVP00Q2) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q3) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q4) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q5) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q6) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q7) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q8) GROUP($GROUP)
DEFINE PROGRAM(BVP00Q9) GROUP($GROUP)
DEFINE PROGRAM(BVP00SA) GROUP($GROUP)
DEFINE PROGRAM(BVP00SI) GROUP($GROUP)
DEFINE PROGRAM(BVP00S1) GROUP($GROUP)
DEFINE PROGRAM(BVP00S3) GROUP($GROUP)
DEFINE PROGRAM(BVP00S4) GROUP($GROUP)
DEFINE PROGRAM(BVP00S5) GROUP($GROUP)
DEFINE PROGRAM(BVP00S6) GROUP($GROUP)
DEFINE PROGRAM(BVP00S7) GROUP($GROUP)
DEFINE PROGRAM(BVP00S8) GROUP($GROUP)
DEFINE PROGRAM(BVP00S9) GROUP($GROUP)
DEFINE PROGRAM(BVP00TA) GROUP($GROUP)
DEFINE PROGRAM(BVP00TT) GROUP($GROUP)
DEFINE PROGRAM(BVP00TU) GROUP($GROUP)
DEFINE PROGRAM(BVP00TV) GROUP($GROUP)
DEFINE PROGRAM(BVP00TW) GROUP($GROUP)
DEFINE PROGRAM(BVP00TX) GROUP($GROUP)
DEFINE PROGRAM(BVP00TY) GROUP($GROUP)
DEFINE PROGRAM(BVP00TZ) GROUP($GROUP)
DEFINE PROGRAM(BVP00T1) GROUP($GROUP)
DEFINE PROGRAM(BVP00T2) GROUP($GROUP)
DEFINE PROGRAM(BVP00T3) GROUP($GROUP)
DEFINE PROGRAM(BVP00T4) GROUP($GROUP)
DEFINE PROGRAM(BVP00T5) GROUP($GROUP)
DEFINE PROGRAM(BVP00T6) GROUP($GROUP)
DEFINE PROGRAM(BVP00T7) GROUP($GROUP)
DEFINE PROGRAM(BVP00T8) GROUP($GROUP)
DEFINE PROGRAM(BVP00T9) GROUP($GROUP)
DEFINE PROGRAM(BVPRACF) GROUP($GROUP)
DEFINE PROGRAM(BVPR980) GROUP($GROUP)
                        RESIDENT(YES)
DEFINE PROGRAM(BVPR990) GROUP($GROUP)
                        RESIDENT(YES)
DEFINE PROGRAM(BVPSECT) GROUP($GROUP)
DEFINE PROGRAM(BVPUCTR) GROUP($GROUP)
                        RESIDENT(YES)
DEFINE PROGRAM(BVPUCTX) GROUP($GROUP)
*********************************************************************
*   CSD CICS : VA PAC  DSMS   TRANSACTION                           *
*********************************************************************
DEFINE TRANSACTION($BAS.$LNG) GROUP($GROUT)
DESCRIPTION(PACBASE DSMS TRANSACTION) PROGRAM(BVPMONI)
ADD GROUP($GROUP) LIST($LIST)
ADD GROUP($GROUF) LIST($LIST)
ADD GROUP($GROUT) LIST($LIST)
//
```

System parameters PDS allocation/loading

DI1SY module: '$prfj.1SY' job

Caution: change :/ characters with ./ before executing the job.

Step	Program	Comment
DELETE	IDCAMS	DELETE of parameters file
BR14	IEFBR14	Reservation of parameters file
UPD	IEBUPDTE	Loading of file members

See the description in Subchapter 'System parameter library', Chapter 'DSMS components'.

Execution JCL

```
//$PRFJ.1SY  JOB ($CCPT),'SY PDS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*                                                                   *
//*    ALLOCATION OF VA PAC DSMS PARAMETER PDS                        *
//*     .STEP1 : LISTCAT                                              *
//*     .STEP2 : ALLOCATION                                           *
//*                                                                   *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*  - REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*  - THIS PDS CONTAINS THE SYSINS FOR ALLOCATING THE FILES THAT     *
//*  MAKE UP THE PACBASE DSMS FILES                                   *
//*  THE INDICATED SIZES CAN BE ADAPTED ACCORDING TO YOUR NEEDS       *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  LISTC ENT($INDSN..BVPSY)
/*
//STEP2    EXEC PGM=IEFBR14,COND=(0,EQ,STEP1)
//SY       DD DSN=$INDSN..BVPSY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),VOL=SER=$VOLSN,
//         SPACE=(6080,(50,,5)),UNIT=$UNITSN
//*
//*********************************************************************
//*        --- LOADING OF VA PAC DSMS PARAMETERS IN 'SY' PDS          *
//*********************************************************************
//UPD    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..BVPSY,DISP=SHR
//SYSIN    DD *
:/     ADD NAME=DF$BAS.0DA
```

```
 DELETE ($INDUV..$BAS.0DA) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BAS.0DA) -
                  SHR (2,3) REUSE  KEYS (40,2)        -
                  VOL ($VOLUV)  CYL (5,1)           -
                  RECSZ (80,350) )                   -
        INDEX   ( NAME ($INDUV..$BAS.0DA.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDUV..$BAS.0DA.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$BAS.0DC
 DELETE ($INDUV..$BAS.0DC) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BAS.0DC) -
                  SHR (2,3) REUSE  KEYS (31,2)        -
                  VOL ($VOLUV)  CYL (1,1)           -
                  RECSZ (50,168) )                   -
        INDEX   ( NAME ($INDUV..$BAS.0DC.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDUV..$BAS.0DC.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$BAS.0DX
 DELETE ($INDUV..$BAS.0DX) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BAS.0DX) -
                  SHR (2,3) REUSE  KEYS (50,0)        -
                  VOL ($VOLUV)  CYL (5,1)           -
                  RECSZ (80,80) )                    -
        INDEX   ( NAME ($INDUV..$BAS.0DX.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDUV..$BAS.0DX.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$BAS.0DJ
 DELETE ($INDUV..$BAS.0DJ) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BAS.0DJ) -
                  SHR (2,3) REUSE  NUMBERED           -
                  VOL ($VOLUV)  CYL (1,1)           -
                  RECSZ (180,180) )                  -
        DATA    ( NAME ($INDUV..$BAS.0DJ.D) -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPDE
 DELETE ($INDSV..BVPDE) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPDE) -
                  SHR (2,3) REUSE  KEYS (17,0)        -
                  VOL ($VOLSV)  CYL (7,1)           -
                  RECSZ (90,90) )                    -
        INDEX   ( NAME ($INDSV..BVPDE.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDSV..BVPDE.D) -
               CISZ (4096) )     /*: CATALOG ($SCAT) */
:/     ADD NAME=DF$BAS.0DF
 DELETE ($INDSV..$BAS.0DF) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..$BAS.0DF) -
                  SHR (2,3) REUSE  KEYS (37,2)        -
                  VOL ($VOLUV)  CYL (5,1)             -
```

```
                  RECSZ (100,554) )                    -
          INDEX   ( NAME ($INDSV..$BAS.0DF.I) -
                    CISZ (4096) )                      -
          DATA    ( NAME ($INDSV..$BAS.0DF.D) -
                    FSPC (10,5)                        -
                 CISZ (4096) )     /*: CATALOG ($SCAT) */
:/        ADD NAME=VERIFDA
 VERIFY FILE (PACDDA)
:/        ADD NAME=VERIFDC
 VERIFY FILE (PACDDC)
:/        ADD NAME=VERIFDX
 VERIFY FILE (PACDDX)
:/        ADD NAME=VERIFDJ
 VERIFY FILE (PACDDJ)
:/        ADD NAME=VERIFDE
 VERIFY FILE (PACDDE)
:/        ADD NAME=LI$BAS.0DJ
 LISTCAT ENTRIES ($INDUV..$BAS.0DJ) CLUSTER
:/     ADD NAME=MAXKEY
999999999999
:/     ADD NAME=REPRO999
 REPRO INFILE (MAXKEY)   OUTFILE (SYSPAF)
/*
//
```

GDG allocation/initialization

D12PRE module: '$prfj.2PR' job

Caution: If SMS is installed on your site, delete DD //GDGMOD from the GDGBJ and GDGBB steps before executing the job.

Step	Program	Comment
DELETE	IDCAMS	DELETE of DSCB MODEL file
BR14	IEFBR14	Reservation of DSCB MODEL file
GDGBJ	IDCAMS	GDG allocation of BJ Journal file
INIBJ	IEBGENER	Initialization of BJ Journal file
GDGBB	IDCAMS	GDG allocation of BB backup file
INIBB	IDCAMS	Copy of backed-up test deck on disk

See the description in Subchapter 'User files', Chapter 'DSMS components'.

Execution JCL

```
//$PRFJ.2PR JOB ($CCPT),'PREPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*   JOB TO RUN ONLY THE FIRST TIME VA PAC DSMS IS INSTALLED         *
```

```
//*    . BUILDING OF DSCB MODEL AND INDEX DATA-GROUP FOR             *
//*      "BB" AND "BJ" SAVE FILES                                    *
//*    . "BJ" FILE INITIALIZATION                                    *
//*    . LOADING OF DSMS TEST FILES ON "BB" FILE                     *
//*  ->NOTE                                                          *
//*    ----                                                          *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS           *
//*****************************************************************
//*
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($DSCB)
//*
//BR14     EXEC PGM=IEFBR14
//DSCB     DD DISP=(,CATLG),UNIT=$UNITSN,VOL=SER=$VOLSN,SPACE=(TRK,0),
//            DSN=$DSCB
//GDGBJ    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD  DSN=$INDUN..$BAS.0BJ,
//        DISP=(,KEEP,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,0),
//        DCB=($DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD *
  DEFINE GENERATIONDATAGROUP -
           (NAME ($INDUN..$BAS.0BJ) LIMIT (3) SCR)
//*
//INIBJ     EXEC PGM=IEBGENER
//SYSIN    DD  DUMMY
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DUMMY,DCB=(RECFM=FB,LRECL=180,BLKSIZE=180)
//SYSUT2   DD  DSN=$INDUN..$BAS.0BJ(+1),
//        DISP=(,CATLG,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,1),
//        DCB=($DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//*
//GDGBB    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD  DSN=$INDUN..$BAS.0BB,
//        DISP=(,KEEP,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,0),
//        DCB=($DSCB,RECFM=VB,LRECL=354,BLKSIZE=6376)
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD *
  DEFINE GENERATIONDATAGROUP -
           (NAME ($INDUN..$BAS.0BB) LIMIT (3) SCR)
//INIBB   EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//BBO  DD  DSN=$INDUN..$BAS.0BB(+1),DISP=(,CATLG,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,(500,10),RLSE),
//        DCB=($DSCB,RECFM=VB,LRECL=354,BLKSIZE=6376)
```

```
//BBI      DD  DSN=$HLQ..SBVPDF2(BVPBB),DISP=SHR
//SYSIN    DD  *
  REPRO INFILE (BBI)   OUTFILE (BBO)
//
```

Loading of error messages

DI4DE module: '$prfj.4DE' job

This job is used to define and load the error messages and help function file and the DAF work file.

Step	Program	Comment
DELDEF	IDCAMS	DELETE/DEFINE of the DE and DF files
REPRO	IDCAMS	REPRO of the DE file

See the description in Subchapter 'System files', Chapter: 'DSMS components'.

Execution JCL

```
//$PRFJ.4DE  JOB ($CCPT),'DSMS DE FILE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*    LOADING OF VA PAC-DSMS ERROR MESSAGE FILE                      *
//*    AND PAF WORK FILE                                              *
//*********************************************************************
//DELDEF   EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//*****  DELETE DEFINE 'DE' FILE
//*
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD DSN=$INDSN..BVPSY(DFBVPDE),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DF$BAS.0DF),
//      DISP=SHR
//REPRO    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//*****  LOADING 'DE' FILE
//*
//SYSPRINT DD SYSOUT=$OUT
//DEO      DD  DSN=$INDSV..BVPDE,DISP=SHR
//DEI      DD  DSN=$HLQ..SBVPDF5(BVPDE),DISP=SHR
//SYSIN    DD  *
  REPRO INFILE (DEI)   OUTFILE (DEO)
//*
//REPRO    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//*
//SYSPRINT DD SYSOUT=$OUT
//DFO      DD  DSN=$INDSV..$BAS.0DF,DISP=SHR
//DFI      DD  *
99999999999999999999999999999999999999999
```

```
/*
//SYSIN    DD  *
  REPRO INFILE (DFI)   OUTFILE (DFO)
//*
```

Loading of batch procedures

It is recommended that all operation procedures be cataloged in one procedures library:

- Either in a reserved PROCLIB: in this case, execute the allocation job first, and then the loading job.
- Or in an existing PROCLIB: in this case, execute the loading job straight away.

DI5IPRO module: '$prfj.5PI' job

Allocation of a reserved library (optional)

Step	Program	Comments
STEP1	IEFBR14	Allocation of procedures library

Execution JCL

```
//$PRFJ.5PI  JOB ($CCPT),'PAC DI5IPRO',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*               INSTALLATION - DI5IPRO                              *
//*                                                                   *
//*                WARNING! OPTIONAL JOB                              *
//*                ====================                               *
//*                                                                   *
//*     INITIAL ALLOCATION OF A SPECIAL "PROCLIB" FOR THE PRODUCT     *
//*     .STEP1 : LISTCAT                                              *
//*     .STEP2 : ALLOCATION                                           *
//*                                                                   *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  LISTC ENT($BIBP)
/*
//STEP2  EXEC PGM=IEFBR14,COND=(0,EQ,STEP1)
//LIB      DD DSN=$BIBP,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(200,20,10))
```

DI5PRO module: '$prfj.5PR' job

Loading of procedures

This job includes the IEBUPDTE step, which creates one member for each procedure.

Caution:

Replace all :/ with ./ before submitting the job.

Each member is coded 'BVPNNNN', where NNNN is the standard name of the procedure.

Procedures are detailed in the other chapters of this manual.

Execution JCL

```
//$PRFJ.5PR JOB ($CCPT),'PROCEDURES ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*       ---- CATALOGING PROCEDURES ----                            *
//*            ====================                                   *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
//  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2 DD DSN=$BIBP,DISP=SHR
//SYSIN   DD   DATA,DLM='F+'
:/ ADD NAME=BVPDARC
:/ ADD NAME=BVPDEXP
:/ ADD NAME=BVPDEXQ
:/ ADD NAME=BVPDEXT
:/ ADD NAME=BVPDINI
:/ ADD NAME=BVPDINS
:/ ADD NAME=BVPDLVB
:/ ADD NAME=BVPDPDF
:/ ADD NAME=BVPDPRT
:/ ADD NAME=BVPDREN
:/ ADD NAME=BVPDREO
:/ ADD NAME=BVPDRST
:/ ADD NAME=BVPDSAV
:/ ADD NAME=BVPDUPT
:/ ADD NAME=BVPDXBJ
F+
//
```

Loading of DSMS test database

DI6DS module: '$prfj.6DS' job

DRST : restoration of the test deck files.

See the description in Subchapter 'User files', Chapter 'DSMS components'.

Execution JCL

```
//$PRFJ.6DS   JOB ($CCPT),'TEST FILES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*             - LOADING TEST FILES -                                *
//*********************************************************************
//DRST    EXEC BVPDRST
 R$LNG REC
//
```

CICS changes

Modification of the CICS start-up JCL

The CICS start-up JCL may have to be completed in order to take all the DSMS elements into account:

```
//DFHRPL   On-line load modules library (SBVPMTR8)
```

Modification of transactions output

Upon exiting the xxyE DSMS transaction, the system sends a CICS RETURN via the BVPMOSO program called by XCTL.

If it seems more convenient, you may replace this program with a program of your own, for instance to return control to a user general menu or any other application.

See Subchapter 'CICS Environment', in Chapter 'Environment'.

Complement - User entry-points source files

This complementary installation should only be performed by users who wish to add checks on Change, Event, Query, Report or Site definition screens and by DAF users.

You can download the source files from the VA Pac Support web page:

http://www.ibm.com/software/ad/vapacbase/support.html

1. INSTALLATION OF SOURCES FOR USER CHECKS

Five batch sub-programs (BVPDSCAM BVPDSCEV BVPDSCMQ BVPDSCRQ BVPDSCSI) and 5 on-line sub-programs (BVPCUAM BVPCUEV BVPCUMQ BVPCURQ BVPCUSI) are shipped so that the DSMS Administrator can insert additional controls in them. They must be compiled and linked in the DSMS load-modules libraries.

To make them active, the activation must be specified during restoration. (See the chapter dedicated to the DRST procedure.)

BATCH TRANSACTIONS FOR DAF

These transactions can be used for writing programs which use the DAF facility. They must therefore be updated in the VA Pac database via the UPDT procedure. (See next Subchapter 'Installation of the DAF Environment').

2. INSTALLATION OF THE DAF ENVIRONMENT

The use of the DAF facility implies the transformation of the DSMS Database access SQL requests written in user programs, through the generation of data and calls to Database access sub-programs in the COBOL source generated from these programs.

The PAF Preprocessor processes the generated programs in order to perform this transformation. It includes the BVPDAFP program installed in the batch load-modules library.

To process the generated programs that use DAF, the DPDF procedure is available. It should be used in one of the following ways:

- Request this procedure in the Optional Control Cards in front of/in back of program, which are combined with the link-edit compilation JCL.
- Call this procedure after the execution of the standard GPRT procedure, from which the generated flow will be retrieved.
- Use any other method best suited to the characteristics of the site.

(Refer to the subchapter dedicated to the DPDF procedure.)

Three DAF sub-programs are provided in the installation deck:

- Two batch sub-programs installed in the batch load-modules library:
  - BVPDSBDF for DAF standard requests
  - BVPDSDAC for the physical accesses to the DSMS database

    Note:

    These sub-programs should be transferred to the user program library(ies), either to be included in the "Link-edit" of the user programs (static call), either to be called upon execution (dynamic call).

- One on-line sub-program, BVPTPDF, installed in the on-line modules library.

  The sample JCL of a user batch program which calls DAF is supplied in the JCL PDS (in the JCLDAF member). (See below a sample user batch DAF JCL).

  The work file necessary for the operation of DAF in on-line mode has a CICS-imposed DDNAME, formatted as base.DF. This DDNAME must be unique for all the programs which access the same DSMS Database.

  Data Element, Data Structure and Segment entities used to write programs involving DAF, are provided as batch transactions from the VA Pac Support web page at http://www.ibm.com/software/ad/vapacbase/support.html

IMPORTANT:

Loading the 'DAF dictionary' in the VA Pac database via the UPDT batch update procedure is the responsibility of the Database Administrator, who must make sure that the codes of the entities provided do not conflict with entities that are already defined in the Database.

In order to avoid compatibility conflicts between the site's Dictionary and entities provided for the DAF facility, it is recommended to create an independent library network that will be accessed by the site's DAF utilities. However, this Dictionary may be loaded in the same library as the PAF Dictionary.

Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*                - JCL EXAMPLE -                                    *
//*    EXECUTION OF A USER D.A.F. BATCH PROGRAM                       *
//*********************************************************************
//DAFBATCH PROC ROOTD=$ROOTD, 2 FIRST CHARACTERS TRANSACTION
//      ROOT2=$ROOT2,         3RD CHARACTER TRANSACTION CODE
//      INDSV='$INDSV',       INDEX OF SYSTEM VSAM FILES
//      INDUV='$INDUV',       INDEX OF USER VSAM FILES
//      INDSN='$INDSN',       INDEX OF SYSTEM NON VSAM FILES
//*:    VSAMCAT='$VCAT',      USER VSAM CATALOG
//*:    SYSTCAT='$SCAT',      SYSTEM VSAM CATALOG
//      STEPLIB=,             USER LIBRARY OF LOAD-MODULES
//      OUT=$OUT              OUTPUT CLASS
//*********************************************************************
//MAXKEY  EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&&SYSDAF,DISP=(NEW,KEEP),
//         SPACE=(CYL,(3,3)),
//         LRECL=420,RECORG=KS,KEYOFF=0,KEYLEN=12
```

```
//MAXKEY   DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//WITHDAF  EXEC PGM=------
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&BAS.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&BAS.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..BVPDE,DISP=SHR
//SYSDAF   DD DSN=&&SYSDAF,DISP=(OLD,PASS)
//------   DD DSN=---
//------   DD DSN=---
//------   DD DSN=---
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//         PEND
//DAFBATCH EXEC DAFBATCH
```

List of installed programs

DI9DINS: '$prfj.DIN' job

This list is output by the '$prfj.DINS' job, which executes the INSL procedure. The list contains:

- The list of the batch and on-line programs with their compilation dates,
- The list of the permanent system files DA and DS with their date of creation.

Keep this list in order to send the VA Pac support the installation references in case of System problems.

USER INPUT: none

Return codes:

Code	Meaning
4	Insufficient memory (increase the REGION parameter)
6	No load module in the library
A	Not standard

Contact IBM if one of the following errors occurs:

Code	Meaning
5	Unknown error
7	Input/Output error

Code	Meaning
8	System error
9	Loading error (System error)

Execution JCL

```
//$PRFJ.DIN  JOB ($CCPT),'PROGRAMS DATE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*************************************************************
//*         LIST OF PROGRAM DATES                             *
//*************************************************************
//DINS   EXEC BVPDINS
```

JCL of the procedure

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//*              - LIST OF INSTALLED MODULES -                        *
//*********************************************************************
//BVPDINS  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDUV='$INDUV',   INDEX OF USER VSAM FILES
//  INDSV='$INDSV',   INDEX OF SYSTEM VSAM FILES
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//  BIBBA='$HLQ..SBVPMBR8', LBRY OF INSTAL. BATCH MODULES
//  BIBTP='$HLQ..SBVPMTR8',  LBRY OF INSTALLED TP MODULES
//  OUT='$OUT',       OUTPUT CLASS
//  UWK='$UWK'        UNIT OF WORK FILES
//*************************************************************
//*   INPUT:
//*   - 1ST LINE: 2 FIRST CHARACTERS TRANSACTION CODE (COL.3)
//*   - OTHER LINES IF SELECTION OF PROGRAMS IS NEEDED:
//*     ONE LINE PER PROGRAM: PROGRAM CODE (COL.3, LENGTH 6)
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&PAC7MB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//PDSXDT EXEC  PGM=BVPDSXDT
//*-----------------------
//STEPLIB  DD  DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD  SYSOUT=&OUT
//PACDDA   DD  DSN=&INDUV..&BAS.0DA,DISP=SHR
//PACDDE   DD  DSN=&INDSV..BVPDE,DISP=SHR
//PACDDS   DD  SYSOUT=&OUT
```

```
//PACDMB   DD  DSN=&&PAC7MB,DISP=(OLD,PASS)
//DFHRPL   DD DSN=&BIBBA,DISP=SHR
//         DD DSN=&BIBTP,DISP=SHR
//SYSUDUMP DD  SYSOUT=&OUT
```

Setting of a VA Pac database under DSMS control

To put a VA Pac Database under DSMS control, open the security browser in Administrator workbench.

Then in the 'DSMS Control' tab, associate the DSMS transaction code ('DSMS Database code') with the code of the selected VA Pac Database. You can associate one DSMS Database code with several VA Pac codes, or associate one DSMS Database code with one VA Pac code.

An 'Administrator' Profile is required for this operation.

ADDITIONAL USER OPERATIONS

The DSMS function includes a functionality whereby VA Pac must consult the DSMS VA Pac Elements file (DC).

If the DSMS Function is in the same CICS as the VA Pac Database using it, no additional user operation is necessary.

If the DSMS Function and the VA Pac Database are not in the same CICS, the DC file must be declared in Read Only mode (READ, BROWSE) under the CICS in which the VA Pac Database is installed.

If the DSMS function and the VA Pac Database are located in different sites, a copy of the DC file must be installed in the VA Pac site (via an IDCAMS, DELETE/DEFINE, and REPRO). This copy will have to be "refreshed" regularly according to the updates made on the DC file.

These operations are the user's responsibility, as no JCL is provided.

Use tests

These tests include three steps:

- On-line use tests of the Database,
- Extraction utility test,
- Database management tests.

1. ON-LINE USE TESTS

Open the test Database files in on-line mode.

Log in with the user 'TEST'. Use password 'IBM'.

Perform screen branchings and updates.

2. EXTRACTION TEST

Run the '$prfjDEXT' job (DEXT procedure). This job extracts elements from the test Database.

For this test, the Database files can remain open in on-line mode.

3. DATABASE MANAGEMENT TESTS

The purpose of these tests is to execute the Database management procedures.

The following steps must be performed in the indicated order:

- Archiving the journal created during the use tests: Run the '$prfjDARC' job, creates a BJ(1) file.
- Direct backup of the Database: Run the '$prfjDSAV' job, creates a BB(1) file.
- Database restoration from BJ(1) archive and BB(1) Database backup: Run the '$prfjDRST' job.

During all these tests, the Database files must be closed to on-line access.

After the Database is restored, open the Database files and perform another set of quick operational on-line tests.

Test JCL: DEXT

```
//$PRFJ.DXT  JOB ($CCPT),'EXTRACTION',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* EXTRACTION TESTS - EXTRACTION OF USER TABLES                      *
//*                    EXTRACTION OF A REQUEST                        *
//*********************************************************************
//DEXT   EXEC BVPDEXT
 *USER    CGI
 TUD
 TUG
 TUS
 TUP
 QC LISTE
 QE LCHECK       $/19931010/
//PDSEX.PACDIM DD SYSOUT=$OUT
//*
//*     PDSEX.PACDIM DD DSN=&&PACDIM,DISP=(,PASS),
//*            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//*            DCB=(RECFM=FB,LRECL=250,BLKSIZE=5000)
```

```
//*     DUPT   EXEC BVPDUPT
//*     PDSUP0.PACDIM DD DSN=&&PACDIM,DISP=(OLD,PASS),
//*            DCB=BLKSIZE=5000
//
```

Test JCL: DARC

```
//$PRFJ.DAR  JOB ($CCPT),'DARC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*****************************************************************
//*        TESTING THE JOURNAL ARCHIVAL PROCEDURE                 *
//*****************************************************************
//DARC   EXEC BVPDARC
```

Test JCL: DSAV

```
//$PRFJ.DSA  JOB ($CCPT),'DSAV',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*************************************************************
//*         TESTING THE DATABASE BACKUP                       *
//*************************************************************
//DSAV   EXEC BVPDSAV
```

Test JCL: DRST

```
//$PRFJ.DRS  JOB ($CCPT),'DRST',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*****************************************************************
//*           TESTING THE RESTORATION WITH THE JOURNAL            *
//*                                                               *
//* AFTER THE RESTORATION, MAKE A FEW QUICK TESTS FOR ON-LINE     *
//* FUNCTIONNING, AFTER HAVING RE-OPENED THE DATABASE FILES.      *
//*****************************************************************
//DRST   EXEC BVPDRST
 R$LNG REC
//*
```

Test JCL: DXBJ

```
//$PRFJ.DXB  JOB ($CCPT),'DXBJ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*************************************************************
//* TEST OF THE DSMS JOURNAL EXTRACTION                       *
//*   SELECTION OF DATE, HOUR AND USER                        *
//*************************************************************
//DXBJ   EXEC BVPDXBJ
 *USER    CGI
 K 19940701199407150000000240000USER
//PDS700.PACDIM DD SYSOUT=$OUT
//*
//*PDS700.PACDIM DD DSN=&&PACDIM,DISP=(,PASS),
//*       UNIT=&UWK,SPACE=(TRK,(15,5),RLSE),
//*       DCB=(RECFM=FB,LRECL=250,BLKSIZE=5000)
```

```
//*DUPT   EXEC BVPDUPT
//*PDSUP0.PACDIM DD DSN=&&PACDIM,DISP=(OLD,PASS),
//*       DCB=BLKSIZE=5000
//
```

Standard reinstallation

DSMS must be re-installed when a new version of the software comes out with corrections and enhancements.

To install this version, download the cartridge in the dedicated PDS by using SMP/E and execute the JCL supplied if it is necessary.

This version is identified by a number and includes:

- An installation cartridge (or tape),
- The "Program Directory for DSMS" specific to SMP/E,
- The list of corrected anomalies,
- Instructions -- possibly included -- to complete this chapter.

Generally, only system files and program libraries are impacted by this version.

In any case, load-modules are updated by SMP/E. They are copied to hlq.SBVPMBR8 and hlq.SBVPMTR8 PDS.

Three cases are possible:

Case 1: The installation JCLs have been kept

Case 2: The installation JCLs must be re-generated and the reinstallation is standard.

Case 3: The installation JCLs must be re-generated and the reinstallation is not standard.

CASE 1: THE INSTALLATION JCLs HAVE BEEN KEPT

For a STANDARD REINSTALLATION, run the job contained in the following JCL:

```
- DI4DE:  $prfj.4DE job for error messages initialization.
```

CASE 2: YOU MUST RE-GENERATE INSTALLATION JCLs FOR A STANDARD REINSTALLATION

To obtain more details about the procedures to perform, refer to Subchapters 'Initial JCL' and 'Complete JCL Installation'.

To re-generate JCLs, run the BVPMMJCL utility again with the parameters set for the site installation and the JCLs needed for the reinstallation.

Check the resulting JCLs. Perform the reinstallation according to the steps of CASE 1.

CASE 3: YOU MUST RE-GENERATE THE INSTALLATION JCLs FOR A NON-STANDARD REINSTALLATION

To get the JCLs, see CASE 2.

Once you have the JCLs, follow the special instructions indicated in the instructions provided with the version.

Retrieving a 2.n version

To retrieve a 2.n version, you must perform the following operations:

- backup the 2.0 version,
- reorganize the 2.0 version,
- restore in the new version's environment, using the file produced by the previous reorganization.

Chapter 19. DLVB - Replacement of low-values with blanks

The DLVB procedure inserts a blank wherever a low-value is present in the BB Database backup file.

The purpose of this procedure is to make possible the transfer of the BB file onto various platforms, while avoiding problems due to the presence of low-values during these transfers.

Utilization option

The DLVB procedure gives the user the opportunity to produce a transfer file containing only the 'data'-type records (refer to next subchapter).

In this case, the backup file obtained on the target platform after transfer will have to be reorganized (DREO procedure) in order to rebuild the cross-references file (DX file).

Execution conditions

None.

DLVB - Parameters / Description of steps

Replacement of low-values with blanks: PDSLVB

On the EXEC line, specify PARM='DATA' to keep only the DATA records in the output file.

To keep both the INDEX and DATA records, do not specify anything.

Code	Physical name	Type	Label
PACDBB	&INDUN..&base.BB(0)	Input	Database backup
PACDB1	&INDUN..&base.BB(+1)	Output	New Database backup

DLVB - Execution JCL

```
//*********************************************************************
//* DSMS                                                              *
//*                                                                   *
//* - REPLACEMENT OF LOW VALUE CHARACTERS WITH BLANK CHARACTERS -     *
//*                                                                   *
```

```
//*  OPTION : SUBMIT PROCEDURE WITH PARM='DATA' TO PROCESS DATA O     *
//*********************************************************************
//BVPDLVB  PROC BAS='$BAS',  CODE OF DSMS DATABASE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//*********************************************************************
//PDSLVB EXEC PGM=BVPDSLVB,PARM=' '
//*--------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDBB   DD DSN=&INDUN..&BAS.0BB(0),DISP=SHR
//PACDB1   DD DSN=&INDUN..&BAS.0BB(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPABB,
//            DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 20. DSMS - RACF or TOPSECRET Interface

Introduction

Security systems provide a mechanism for data access control. They validate user codes and passwords.

The Security Systems Interface is designed to ensure the communication of controls between the security system installed on the site and DSMS.

The Security Interface performs the following DSMS-related tasks:

- In on-line mode: automatic retrieval of the CICS or IMS SIGN-ON USERID, which is displayed on the log-on screen.
- In batch mode: for the DSMS procedures which include user input ('*' line), it will be possible to leave the user code and the password blank.

Installation

OPERATION OF THE DSMS / RACF OR TOPSECRET INTERFACE

- For RACF:

  ON-LINE REQUESTS

  A command enables you to run a RACF request in on-line mode to determine the users' authorizations in relation to the resources ('EXEC QUERY SECURITY' in the BVPRACF control sub-program).

  BATCH REQUESTS

  No batch command enables you to perform a RACF request ; so you must execute the RACF request via a BVPSECUR assembler sub-program, supplied by SMP/E in the hlq.SBVPMBR8 PDS.

  Moreover you will have to install an SVC in the LPA library if this has not already been done upon a VA Pac or Pactables installation.

  This installation will involve the following steps:

  1. Declaration of the SVC with the chosen routine number.

  For example, for the number 232, add, in SYS1.PARMLIB(IEASVC00), the line SVCPARM 232,REPLACE,TYPE(3),EPNAME(IGC0023B) where IGC0023B is the SVC load-module name.

  2. link of the BVPSECUR object module to create the load-module name,

  3. Creation of this load-module name in the SVC load system library,

  4. ZAP of the BVPSECB VA Pac module with the chosen SVC routine number,

5. After updating the LPA library, you will have to perform an IPL CLPA to take these modifications into account.

The E10RACF sample JCL supplied with the installation JCLs deals with the points 2 and 3, and the E11RACF sample JCL deals with the point 4.

If the SVC has already been installed upon a VA Pac installation, you just have to run the E11RACF JCL.

After performing these operations, you must check that the user is authorized to submit the procedures.

- For TOPSECRET

The sources of the sub-programs (assembler) which access the TOPSECRET tables are supplied by SMP/E in the hlq.SBVPSRC PDS and must be compiled.

COMPILATION OF ACCESS SUB-PROGRAMS

BVPTSS and BVPTSSC must be compiled with the ″OPMAT″ TSS library in the SYSLIB of the Assembler compilation.

BVPTSSC is a CICS program and must be translated before being compiled and link-edited.

BVPTSSC and the TSSCAI program (Computer Associates) must be declared in the CICS CSD and must be located in one of the DFHRPL's load-module library.

Operating mode

The interface performs its controls according to the 'Security Systems Interface' indicator specified in the DRST procedure.

If this indicator is set to call RACF or TOPSECRET, a user can log on with a user code other than his/her own.

The control of the users is managed by the security interface, and a user can log on only with his/her own code.

- Logging on in on-line mode: the DSMS SIGN-ON screen is initialized with the code under which the user logged on to CICS or IMS. This code is retrieved in the IO-PCB in IMS and by an EXEC CICS ASSIGN USERID command in CICS (valid only from CICS release 1.7 onwards). Changing the user code is prohibited.
- The password field is locked and left blank. The cursor is positioned on the library code.

- LVQ screen (RACF only): since RACF does not carry over the user code and the CICS or IMS password, you have to insert them on the JOB card. Since DSMS reads the password in the TUD table, the password must be identical to that declared in CICS or IMS.

1. Batch procedures that include a '*' line: the user code and password are no longer required since the system automatically takes the code under which the user signed on to TSO.

   As a result, the PASSWORD is no longer present in the temporary files found in batch job streams, including DPRT, since the security system ensures the propagation of the USER code and the PASSWORD.

   Another consequence is that job streams including steps with a '*' line can be linked together without manual intervention so that the password can be specified.

   For RACF, this process implies a restriction: the user cannot code several '*' lines with user codes other than his/her own for procedures which would normally allow the user to do so (such as GPRT).

   For TOPSECRET, the user can never enter user codes other than his/her own.

Part Number: DEDCI000351A - 7345

Printed in USA