VisualAge Pacbase 2.5

VA PAC 2.5 – IBM CICS/OS/MVS
OPERATIONS MANUAL VOLUME I : ENVIRONMENT & INSTALLATION

DEPCI001251A

Note

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

According to your license agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:
http://www.software.ibm.com/ad/vapacbase/support.htm

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

First Edition (April 1998)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:
http://www.software.ibm.com/ad/vapacbase/support.htm
or to the following postal address:
IBM Paris Laboratory
VisualAge Pacbase Support
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

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

NOTICES

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

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

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

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

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

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

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

TRADEMARKS

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

TABLE OF CONTENTS

1. FOREWORD

FOREWORD

HOW TO USE THIS MANUAL

This manual is intended for the person in charge of installing the VisualAge Pacbase system.

It describes the system's COMPONENTS and its ENVIRONMENT, lays out recommendations for the INSTALLATION of the new release, and explains the operations that must be performed for a standard RE-INSTALLATION of correction versions.

USERS OF PREVIOUS SYSTEM RELEASES

It is generally recommended to install the new release in an environment distinct from that of any earlier release, particularly as far as the installation parameters are concerned. To complete the new installation, the set of tests provided on the installation media must be run.

. Retrieval from VisualAge Pacbase releases older than 8.02v02.

Contact VisualAge Pacbase Support Center.

. Retrieval from VisualAge Pacbase 8.02v02, 1.2, 1.5, 1.6, 2.0

In this case, the new release may be installed in the same environment as the older release. In this case, refer to chapter '8.02v02, 1.2, 1.5, 1.6 RETRIEVAL', or chapter '2.0 RETRIEVAL'

2. VISUALAGE PACBASE COMPONENTS

2.1. INTRODUCTION

INTRODUCTION

One of the goals of the VisualAge Pacbase system is to manage permanent data in either batch or on-line mode, by using two types of resources:

LIBRARIES which store the system programs, and the parameters needed to run them:

- One On-Line Program library,
- One Batch Program library,
- One System Parameter library,
- One Parameter library for each VisualAge Pacbase Database
- One library for the batch procedure's JCLs.

PERMANENT FILES, containing the data handled by the system programs. These files can be classified into two categories:

. 'System' files, which are not linked to a particuliar VisualAge Pacbase database and remain relatively unchanged,

. 'Evolving' files, which are associated to a VisualAge Pacbase Database, and whose volumes vary according to the updates performed.

NOTES:

The WorkStation, DSMS, revamped DSMS, Pacbase Web Connection, and Pactables Functions are installed independently of the other VisualAge Pacbase functions.

The VisualAge Pacbase-ENDEVOR Interface must also be installed independently from all other functions.

The installation and operation of these Functions and Facilities are described in the operations manuals specific to each one.

2.2. CODES OF FUNCTIONS, EXTENSIONS AND UTILITIES

CODES OF FUNCTIONS, EXTENSIONS AND UTILITIES

The following lists provide abbreviated codes for system functions, extensions, and optional utilities:

```
.Specifications Dictionary                = DIC

.Extensions:

   -Personalized Documentation Manager    = PDM
   -Security Systems Interface            = SEC

.Optional Utilities:

   -Sub-Network Comparison Utility        = LCU
   -Rename/Move Entity Utility            = RME
   -Journal Statistics Utility            = ACT

.Functions/Facilities:

   -Structured Code                       = SC
   -Batch Systems Development             = BSD
   -COBOL Generator                       = COB
   -On-Line Systems Development           = OSD
   -Client/Server Facility                = OCS
   -DBD                                   = DBD
   -DBD/ Relational SQL                   = SQL
   -Pactables                             = TAB
   -Development & Support Management System
    (DSMS)                                = DSM
   -Production Environment Interface      = PEI
   -Dictionary Extensibility              = DEX
   -Pactransfer                           = TRF
   -VA Smalltalk / VA Pac Bridge          = VIS
   -VA Pac / TeamConnection Bridge        = PTC
   -Impact Analysis                       = S2K
   -Pacbench Quality Control              = PQC
   -The WorkStation                       = WST
   -Pacbase Access Facility               = PAF
   -PacReverse                            = REV
   -Pacbase Web Connection                = PAW
```

2.3. ON-LINE PROGRAM LIBRARY

THE ON-LINE PROGRAM LIBRARY (MTR8)

Its size is approximately 1,500 blocks of 6,144 bytes.

PROGRAM CODE	FUNCTION OPTION	CORRESPONDING CHOICE COMMENTS
		xxEE transaction
xxPA00	DIC	.Connection screen
xxPA01	-	H
xxPA10	-	HP .
xxPA11	-	LCPC..
xxPA12	-	PC..
xxPA13	-	PT.
xxPA14	-	PE.
xxPA15	-	PU........
xxPA16	-	PK
xxPA17	-	PD
xxPA18	-	LCPU........
xxPA19	-	PW
xxPA20	-	PB
xxPA21	-	PM
xxPA22	-	LCPM
xxPA30	PEI	Prod. Env. Interface menu
xxPA31	-	EE.....
xxPA32	-	EG............
xxPA33	-	ES....
xxPA34	-	LSEP..............
xxPA35	-	ED........
xxPBND	DIC	Abend map
xxPPHP	-	Help
		VA Pacbase transaction
xxQA00	DIC	D..
xxQB00	BSD	R...
xxQC00	DIC	E......
xxQC01	-	LUE
xxQC50	WST	++5 Up/Dw mapping
xxQD00	SC	P......B et O......B
xxQE00	DIC	E......D
xxQF00	COB	P......SC
xxQF10	-	P......STR
xxQG00	DIC	K......
xxQH00	OSD	O......
xxQH01	DIC	LA LC LE LF LM LN LP LS LT LX
xxQH20	OSD	O......CS
xxQH30	-	O......O
xxQI00	-	O......L

PROGRAM CODE	FUNCTION OPTION	CORRESPONDING CHOICE COMMENTS
xxQI01	OSD	O......CE (C1)
xxQI02	-	O......CE (C2)
xxQI03	-	O......SIM
xxQI04	-	O......ADR
xxQI05	-	O......CE (C3)
xxQI20	-	O......M
xxQI21	-	
xxQI50	WST	++4 Up/Dw screen data element
xxQK10	DIC	M......
xxQK20	-	M......CM
xxQK30	-	M......CE
xxQL10	-	B......
xxQL20	-	B......DH
xxQL21	-	B......DT
xxQL30	-	B......DC
xxQL40	SQL	B......DR...
xxQL41	-	B......DR...K
xxQL45	-	B......GEN
xxQL46	-	
xxQM00	SC	P......CP et O......CP
xxQP00	-	P......P et O......P
xxQP01	-	Display P......TC
xxQP02	-	Display O......TC
xxQP03	-	P......TC et O......TC
xxQP04	-	P......TO
xxQP05	-	O......TO
xxQP06	-	P......PG et O......PG
xxQP07	-	O......PG
xxQP08	-	P......PG
xxQP50	WST	++6 Up/Dw specific codes
xxQR00	DIC	LL........L_......
xxQS02	-	-XP
xxQS03	-	-ACT
xxQS04	-	WS
xxQS05	-	?
xxQS06	-	Menus
xxQS08	DEX	-XQ
xxQT00	DIC	T......D
xxQT10	-	T......
xxQT20	PDM	T......SIM
xxQT50	WST	++2 Up/down textes
		xxEE transaction
xxQUPA	DIC	English monitor
xxQUPF	-	French monitor

PROGRAM CODE	FUNCTION OPTION	CORRESPONDING CHOICE COMMENTS
		VA Pacbase transaction
xxQU00	DIC	U..
xxQU01	-	U......D
xxQU10	PDM	V......
xxQU20	-	V......D
xxQV10	DIC	I......
xxQV20	-	I......D
xxQV30	-	-G
xxQX00	-	*
xxQX01	-	LH
xxQY01	DEX	F......
xxQY02	-	F......CE
xxQY03	-	$ ....
xxQY04	-	$ ....D
xxQY05	-	Q......
xxQY10	WST	++1 Design entities Upload
xxQY11	-	++3 Design entities Download
xxQY20	DIC	GP
xxQY30	-	JO
xxQZ00	-	VA Pac initial screen
xxQ000	SC	P......
xxQ100	-	P......CD
xxQ101	-	P......HCD
xxQ102	-	
xxQ103	-	
xxQ104	-	
xxQ200	DIC	S....
xxQ210	TAB	S....SS
xxQ300	DIC	S....CE
xxQ400	BSD	R...L
xxQ500	-	R...D
xxQ600	-	R...CE
xxQ700	SC	P......W et O......W
xxQ800	-	P......8
xxQ900	-	P......9
xxR000	DIC	VA Pacbase monitor
xxR005	-	Return to CICS
xxR100	-	Choice processing
xxR200	-	Format conversion
xxR400	-	N*... ou NH....
xxR500	-	Abend map

PROGRAM CODE	FUNCTION OPTION	CORRESPONDING CHOICE COMMENTS
		VA Pacbase transaction
xxR600	DIC	Text editing
xxR980	-	Screen formatting
xxR990	-	WEB formatting
xxUCTR	-	UCTRAN management (ASM) up to CICS ESA 3.1
xxUCTX	-	Neutralization of xxUCTR (COB)
xxUCTZ	-	UCTRAN management (COB) from CICS ESA 3.3
xxSECT	SEC	Security system interface
		PUF-TP transaction
xxFANM	DIC	
xxFBIB	-	
xxFCTL	-	
xxFDBD	-	
xxFECR	-	
xxFENU	-	
xxFFOG	-	
xxFMCL	-	
xxFPGM	-	
xxFRUB	-	
xxFSDO	-	
xxFTXT	-	
xxFVER	-	
xxF000	-	
Sub-programs called in the generated programs :		
CALL	OSD	For I.M.S. OLSD
PBLTDLI	-	IMS or CICS DL/1 OLSD
xxTPST	PAF	Online PAF
xxTPWS	-	Keywords on-line PAF
(must be inserted into the application program library)		

Notes :
xx = program prefix (ROOT, = first two characters of the selected transaction code).

2.4. BATCH PROGRAM LIBRARY

THE BATCH PROGRAM LIBRARY (MBR8)

Its size is approximately 2,000 blocks of 6,144 bytes.

CODE	PROCEDURES	OPTION	COMMENTS
PACA05	UPDT	DIC	
PACA15	UPDT REST	-	
PACA10	GPRT	-	
PACA20	-	-	
PACB	-	-	GPRT monitor
PACB30	-	-	
PACB31	-	SQL	
PACB40	-	DBD	
PACB80	-	-	
PACCTL	PACX	DIC	
PACC30	GPRT	COB	
PACC40	-	-	
PACC80	-	-	
PACDTP	INSL	DIC	
PACD30	-	-	
PACD40	-	-	
PACD80	-	-	
PACD90	-	-	
PACE30	-	OSD	
PACE40	-	-	
PACE80	-	-	
PACFGY	PACX	DIC	
PACFMB	-	-	
PACFTD	-	-	
PACG3C	GPRT	OCS	
PACG3S	-	-	
PACG4S	-	-	
PACG8C	-	-	
PACG8S	-	-	
PACHOI	PACX	DIC	
PACINS	VINS	-	
PACK30	GPRT	OCS	
PACK80	-	-	
PACK90	-	-	
PACL30	-	SC	
PACL80	-	-	
PACL90	-	-	
PACL92	EMUP	-	
PACL93	EMLD	-	

CODE	PROCEDURES	OPTION	COMMENTS
PACL95	GPRT	PAW	
PACM30	-	DIC	
PACM80	-	-	
PACN30	-	PDM	
PACN35	-	-	
PACN40	-	-	
PACN50	-	-	
PACN80	-	-	
PACP30	-	SC	
PACP40	-	-	
PACP80	-	-	
PACP82	-	-	
PACQ	PQCA	PQC	PQCA Monitor
PACR01	INPE	PEI	
PACR10	PRPE	-	
PACR20	GPRT	-	
PACR22	SIPE	PEI	
PACR30	HIPE	-	
PACR40	GRPE	-	
PACR60	SVPE	-	
PACR61	RSPE	-	
PACSJO	PACX	DIC	
PACSMD	-	-	
PACSPU	-	-	
PACSRM	-	-	
PACS30	-	-	
PACS40	-	-	
PACS50	-	-	
PACS60	-	-	
PACS75	-	-	
PACS80	-	-	
PACTIN	GETI	TAB	
PACTI1	GET0	-	
PACT40	GETA GETD	-	
PACT41	GET1 GET2	-	
PACT45	GETA GETD	-	
PACT50	GETD	-	
PACT51	GET2	-	
PACU15	PARM	DIC	
PACU80	-	-	
PACU99	CRYP	-	
PACX	PACX	-	
PADM10	SADM	WST	SSADM integrity check
PAFP10	PPAF GPRP	PAF	PAF pre-processor
PAF900	UPDP	-	PAF update input

CODE	PROCEDURES	OPTION	COMMENTS
PAN200	INFQ	S2K	
PAN205	INFP	-	
PAN210	ISEP	-	
PAN212	ISOS	-	
PAN215	ISEP IANA	-	
PAN220	IPFQ IANA	-	
	IPEP	-	
PAN230	IANA	-	
PAN240	IPFQ	-	
PAN250	IANA	-	
PAN255	IGRA	-	
PAN260	IANA	-	
PAN270	IPIA	-	
PAN280	IPIA	-	
PBBTST	Sub-pgm	PAF	Called by batch user pgms
PBBTWS	-	-	- - - - - - - - -
PBBT98	-	-	Called by PBBTST PBBTWS
PDS600	DEXP	DSM	DEXP before 2.0
PDS610	-	-	- -
PREI00	RVDE	REV	PACREVERSE Interface
PREI40	RVKE	-	- -
PREI50	-	-	- -
PRE986	RVDE	-	- -
PTARSD	RMTD	TAB	Migration: rest. TD
PTAR20	RPTD	-	Migration: retrieval TD
PTASVD	SMTD	-	Migration: saves TD
PTC010	TCLS	PTC	VA Pac-TeamConnection
PTC030	-	-	- -
PTC100	TCGP	-	- -
PTC200	TCME	-	- -
PTC220	TCCI TCME	-	- -
PTC400	TCCI	-	- -
PTC440	-	-	- -
PTED30	XPDM	PDM	PDM Extension
PTED60	-	-	- -
PTEP90	PRGS	-	- -
PTEXD0	XPAF	PAF	PAF Extension
PTEX30	-	-	- -
PTEX80	-	-	- -
PTUBAS	SAVE UPDT	DIC	Database integrity check
PTUCSS	CSES	-	Compression Session No.
PTUESS	ESES	-	Extraction Session No.

CODE	PROCEDURES	OPTION	COMMENTS
PTUG05	TRJC	TRF	Pac/Transfer Facility
PTUG06	-	-	- -
PTUG07	-	-	- -
PTUG10	TRUP	-	- -
PTUG11	-	-	- -
PTUG12	-	-	- -
PTUG42	TRDU	-	- -
PTUG44	-	-	- -
PTUG46	-	-	- -
PTUG50	TRPF	-	- -
PTUG60	TRRP	-	- -
PTUG61	-	-	- -
PTUG90	TRRT	-	- -
PTULOI	RTLO	DIC	Locks retrieval
PTULVB	LVBL	-	Repl. low-values with blanks
PTUQ10	PQCE	PQC	
PTUQ15	-	-	
PTUQ20	PQCA	-	
PTUQ24	-	-	
PTUQ25	-	-	
PTUQ30	-	-	
PTUR00	STOP	-	
PTU001		DIC	All procedures with input transactions
PTU004	REST REAG	-	User code check
PTU100	MLIB	DIC	
PTU120	-	-	
PTU130	SASN	LCU	
PTU140	-	-	
PTU2CL	REOR	DIC	
PTU200	-	-	
PTU208	-	-	
PTU210	-	-	
PTU220	-	-	
PTU240	-	-	
PTU300	ARCH	DIC	
PTU320	-	-	
PTU380	REST	-	
PTU400	-	-	
PTU402	RESY	-	
PTU420	REST	-	
PTU500	SAVE	-	
PTU502	SASY	-	

CODE	PROCEDURES	OPTION	COMMENTS
PTU550	SVAG	DIC	
PTU560	REAG	-	
PTU630	ACTI	ACT	
PTU640	-	-	
PTU810	EMSN	LCU	
PTU815	MESN	-	
PTU850	CPSN	-	
PTU855	-	-	
PVA100	VDWN	VIS	
PVA110	-	-	
PVA300	VUP1	-	
PVA310	-	-	
PVA320	VUP2	-	
PVA400	VPUR	-	
PYSMCC	YSMC	WST	YSM consistency check
PYSMC2	-	-	
PYSMC3	-	-	
UTIXSR	UXSR	DIC	
PACSECB	SUB-PROG	SEC	Called by VA Pac programs
PACN25	GPRT	PDM	- - -
PACN90	-	-	- - -
PACSEP	GPRT UPDT REST	DIC	Banner print sub-program (to separate reports)
PACA90	GPRT UPDT REST	-	Data Element format Analysis sub-program
Retrieval from Pacbase 802.02, 1.2, 1.5, 1.6			
PACR90	PP16	PEI	PEI backup
REP2PJ	PJ16	DIC	Archive
Retrieval from VA Pacbase 2.0			
PTU908	RPPG	DIC	Generation-printing req.

SECURITY SYSTEM(S) INTERFACE EXTENSION (SEC)

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

For RACF, the sub-program coded PACSECU8 must be installed in an authorized library by copying the module which is found in the batch load-module library (PAC.MBR8).

PROGR.	Renamed	Option	Security system
PACSECRA	PACSECU8	DEC	RACF
PACTSS		-	TOPSECRET (Batch)
PACTSSC		-	TOPSECRET (CICS)

For more information on the operation of this extension, refer to Chapter 'Installation', Sub-chapter 'System Complement: Security System Interface', and to the 'Security Systems Interface' Reference Manual.

2.5. VA PAC SYSTEM PARAMETER LIBRARY

THE VA PAC SYSTEM PARAMETER LIBRARY (SY)

It contains all parameters for the utilities used in the batch installation and operation procedures, as well as in the PAF standard user procedures.

```
    .Size:         Approximately 25 blocks of 6,080 bytes
    .Organisation: PDS
    .DCB:          Recfm=FB,Lrecl=80,Blksize=6080
    .DSNAME:       &INDSN..&ROOT.&ROOT.SY
```

Information relating to names, disks (catalog) etc. is initialised according to the initial installation parameters.

CAUTION: THIS INFORMATION SHOULD NOT BE MODIFIED, EXCEPT ACCORDING TO THE NOTES BELOW.

MEMBER	CONTENTS / FORMAT	NOTES
DFxxxxAE	DELETE/DEFINE AE	EDITABLE (1) PARM proc.
DFxxxxAP	DELETE/DEFINE AP	EDITABLE - -
DFxxxxSC	DELETE/DEFINE SC	
DFxxxxSG	DELETE/DEFINE SG	
DFxxxxSR	DELETE/DEFINE SR	
DFxxxxSS	DELETE/DEFINE SS	
DFxxxxSP	DELETE/DEFINE SP	
DFxxxxTS	DELETE/DEFINE TS	VA Pac/T.Connection Bridge
DFxxxxGS	DELETE/DEFINE GS	EDITABLE (2) GPRT proc.
DFSYSPAF	DEFINE	EDITABLE (3)
DLSYSPAF	DELETE	
DFSYSEXT	DEFINE	EDITABLE (5)
DLSYSEXT	DELETE	
DFSYIANA	DEFINE	EDITABLE
DLSYIANA	DELETE	
DFxxxxPA	DELETE/DEFINE PA	EDITABLE (4)
DFTABTDF	DELETE/DEFINE TD	Pactables: Descriptions
DFDSMSDC	DELETE/DEFINE DC	DSMS: VA Pac Elements (To be used if DSMS is projected but not actually installed.)

(1) This size may vary depending on the 'user parameters' contained in the file (for more information, refer to the appropriate chapter describing each file.)

(2) This size depends on the size of the Extraction Master Paths created by the user.

(3) This size depends on the volume of PAF batch requests made when using the file.

(4) This size depends on the volume of PAF on-line request made when using the file.

(5) This size depends on the volume of deep extractor requests.

MEMBER	CONTENTS / FORMAT	NOTES
LIxxxxAE	LISTCAT of AE	
VERIFAN	VERIFY (PAC7AN)	
VERIFAR	VERIFY (PAC7AR)	
VERIFAJ	VERIFY (PAC7AJ)	
VERIFAG	VERIFY (PAC7AG)	
VERIFAE	VERIFY (PAC7AE)	
VERIFAP	VERIFY (PAC7AP)	
VERIFSC	VERIFY (PAC7SC)	
VERIFSG	VERIFY (PAC7SG)	
VERIFSP	VERIFY (PAC7SP)	
VERIFSR	VERIFY (PAC7SR)	
VERIFSS	VERIFY (PAC7SS)	
VERIFAB	VERIFY (PAC7AB)	
VERIFAC	VERIFY (PAC7AC)	
VERIFTD	VERIFY (PAC7TD)	
VERIFEM	VERIFY (PAC7EM)	
SRTREO1	SORT FIELDS	
SRTREO2	SORT FIELDS	
SRTQREO	SORT FIELDS	
SRTCPSN	SORT FIELDS	
SRTRVKE	SORT FIELDS	
MAXKEY	Max record	
REPRO	IDCAMS input	
REPRO999	IDCAMS input	

2.6. VA PAC DATABASE PARAMETER LIBRARY

THE VA PAC DATABASE PARAMETER LIBRARY

```
. Size         : Approx. 10 blocks of 6,080 Bytes
. Organisation : PDS
. DCB          : Recfm=FB,Lrecl=80,Blksize=6080
. Dsname       : &INDUN..&ROOT.&FILE.SY
```

This library contains the SYSIN of the IDCAMS utility called in the VA Pac database batch operation procedures.

Information relating to the file names and the disks (catalogue), is initialised according to the initial installation parameters.

. DELETE/DEFINEs of the database files:

Their names have the DFxxnnff format (xx=ROOT, nn=FILE, and ff=suffix of the relevant file.)

WARNING

The VA Pac database manager can modify the size of the files that make up the database, according to the changes it undergoes, in these members.

. LISTCAT of VSAM file AJ (VA Pac Journal):

Its name has the LIxxnnAJ format.

Member	Contents or format	Notes
DFxxnnAN	DELETE/DEFINE AN	Installation & REST proc.
DFxxnnAR	DELETE/DEFINE AR	- - - -
DFxxnnAJ	DELETE/DEFINE AJ	Instal. & REST ARCH proc.
DFxxnnAG	DELETE/DEFINE AG	Installation & REAG proc.
DFxxnnEM	DELETE/DEFINE EM	For the user application's error message file
PEI Function only:		
DFxxnnAB	DELETE/DEFINE AB	Installation & RSPE proc.
DFxxnnAC	DELETE/DEFINE AC	- - - -
Pac/Transfer Facility only:		
DFxxnnUV	DELETE/DEFINE UV	Creation of parameter file
DLxxnnJT	DELETE JT	Compressed Journal
DLxxnnTJ	DELETE TJ	- -
Pac/Impact Facility only:		
DFxxnnFP	DELETE/DEFINE FP	INFP procedure
VisualAge Pacbase / VisualAge Smalltalk Bridge only:		
DFxxnnVP	DELETE/DEFINE VP	
DLxxnnVP	DELETE VP	
LIxxnnAJ	LISTCAT of AJ	REST procedure

2.7. BATCH PROCEDURES

THE BATCH PROCEDURES

Procedures associated with batch processing are described in Parts II and III of the Operations Manual (Batch Procedures: Administrator's Guide --for those procedures involving only the Database Administrator-- and User's Guide --for those available to the user).

The documentation of each procedure includes the following:

. General presentation

- Introduction
- Execution conditions
- Corrections in case of an ABEND

. Description of user input, processing and results, as well as recommendations for use.

. Description of steps
- List of permanent and temporary files in use,
- Return codes (if any) produced by each step.

. Command file (JCL lines)

PROCEDURE CLASSIFICATION

Batch procedures are documented in the following manuals:

Batch procedures - Administrator's guide:

1) Database Management utilities.

2) Versioning utilities (PEI and Pactransfer).

3) Manager's utilities.

4) Migrations.

Batch procedures - User's guide:

1) Standard procedures.

2) Personalized extraction and automated documentation.

3) Quality analysis and control.

4) Methodology integrity check.

5) Pactables.

6) Impact Analysis.

7) VisualAge Smalltalk / VisualAge Pacbase Bridge.

Environment and installation:

Previous release retrieval procedures:

. Releases 8.02v02 - 1.6:
- Archive Journal retrieval (PJ16)
- Sequential PEI backup retrieval (RTPE)

. Release 2.0:

- Generation-print Requests file retrieval (RPPG)

THE BATCH PROCEDURE LIBRARY

The procedures must be loaded in a special library called PROCLIB.

This library can be an existing library or one specially created for this purpose.

Its characteristics must be the following:

```
     .Size         : Approx. 200 blocks of 6,080 Bytes
     .Organisation: PDS
     .DCB          : Recfm=FB,Lrecl=80,Blksize=6,080
     .DSNAME       : User-defined
```

2.8. SYSTEM FILES

SYSTEM FILES

Besides the libraries described in the preceding sub-chapters, the VA Pac system includes the following permanent files:

. A file containing the ERROR MESSAGES and the ON-LINE DOCUMENTATION of the VA Pac system: AE0.

```
. Size         : approx. 35,000 records
. Organization : sequential
. Lrecl        : 80
. Blksize      : 12,560
. Utilization  : Batch
. Dsname       : &INDSN..&ROOT.&ROOT.AE0
```

This file is not directly used in normal system operation. It is concatenated with the USER PARAMETERS required by the on-line operation of the system (*), to form one file called AE.

```
. Size         : = AE0 + user parameters
. Organization : VSAM-KSDS
. Recsize      : 80
. CI Size      : 4,096 (data) 1,024 (index)
. Key          : 12 (position 0)
. Utilization  : batch and on-line
. Dsname       : &INDSV..&ROOT.&ROOT.AE
```

. A file containing the USER PARAMETERS required by batch operations only (*): AP.

```
. Size         : user parameters
. Organization : VSAM-KSDS
. Recsize      : 80
. CI Size      : 4,096 (data and index)
. Key          : 7 (position 0)
. Utilization  : batch and TP
. Dsname       : &INDSV..&ROOT.&ROOT.AP
```

(*) The AE file handles the following USER PARAMETERS:

User codes,
VA Pac access keys,
Security System Interface activation,

Activation of blank password check,
Activation of batch procedure execution check,
Text types,
Management of accented character conversion,
WorkStation methodology parameterization,
and the DSMS-VA Pac connection.

The AP file handles the following USER PARAMETERS:
modification of fixed parts of standard error messages,
and control cards necessary for the generation of programs.

All user parameters are managed by a special transaction and a special procedure (PARM). For more information, refer to Chapter 'PARM: USER PARAMETER UPDATE', in the BATCH PROCEDURES: Administrator's Guide.

PARM builds the AE and AP files, and manages:

. A USER PARAMETER backup file: PE.

```
. Size         : all user parameters
. Organization : sequential with generation
. Lrecl        : 80
. Blksize      : 6,160
. Utilization  : Batch
. Dsname       : &INDSN..&ROOT.&ROOT.PE(n)
```

The PE file is the actual backup of ALL THE USER PARAMETERS found in the AE and AP files.

. A skeleton generation file: SC, used by the STRUCTURED CODE and BATCH SYSTEMS DEVELOPMENT functions:

```
.Size         : approximately 45 records
.Organization : VSAM-KSDS

.Recsize      : 3,204
.CI Size      : 3,584 (data) 1,024 (index)
.Key          : 4 (position 0)
.Dsname       : &INDSV..&&ROOT.&ROOT.SC
```

. A skeleton file for generation, SG, used by the ON-LINE SYSTEMS DEVELOPMENT, DATABASE DESCRIPTION, and SPECIFICATIONS DICTIONARY functions:

```
.Size         : approximately 400 records
.Organization : VSAM-KSDS
.Recsize      : 4,605
.CI Size      : 5,120 (data) 1,536 (index)
.Key          : 5 (position 0)
.Utilization  : Batch only.
.Dsname       : &INDSV..&&ROOT.&ROOT.SG
```

. A skeleton file for generation, SR, used by the COBOL GENERATOR function:

```
.Size         : approximately 25 records
.Organization : VSAM-KSDS
.Recsize      : 4,605
.CI Size      : 5,120 (data) 512 (index)
.Key          : 5 (position 0)
.Utilization  : Batch only.
.Dsname       : &INDSV..&&ROOT.&ROOT.SR
```

. A skeleton generation file SP, used by the XPAF function for the generation of extractors:

```
.Size         : approximately 5 records
.Organization : VSAM-KSDS
.Recsize      : 4,605
.CI Size      : 5,120 (data) 512 (index)
.Key          : 5 (position 0)
.Utilization  : Batch only.
.Dsname       : &INDSV..&&ROOT.&ROOT.SP
```

. A skeleton generation file SF, used by the XPAF function for the generation of extractors:

```
.Size         : approximately 3,000 records
.Organization : sequential
.Lrecl        : 119
.Utilization  : Batch only
.Dsname       : &INDSV..&&ROOT.&ROOT.SF
```

NOTE: for the generation of COBOL/VS extractors, replace the standard SF COBOL II skeleton file with a COBOL/VS skeleton file with the SF1 suffix.

. A skeleton file for generation, SS, used for the CLIENT/SERVER ON-LINE SYSTEM DEVELOPMENT functions:

```
.Size         : approximately 520 records
.Organization : VSAM-KSDS
.Recsize      : 4,605
.CI Size      : 5,120 (data) 1,536 (index)
.Key          : 5 (position 0)
.Utilization  : Batch only
.Dsname       : &INDSV..&&ROOT.&ROOT.SS
```

2.9. EVOLVING FILES

2.9.1. VISUALAGE PACBASE DATABASE FILES

DATABASE FILES

The first four files make up the actual VA Pac Database. They contain all data related to application development, as follows:

.THE DATA FILE (AR)

```
     .Organization : VSAM-RRDS
     .Recsize      : 140
     .C.I. size    : 2,048
     .Utilization  : Batch and on-line
     .Dsname       : &INDUV..&ROOT.&FILE.AR
     .Size         : 14 records per C.I. of 2,048
```

Each VA Pac line is stored in the data file under a fixed internal number.

The successive states of a given line from the various archived sessions form a chain; at the top of this chain is the most recent state of the line and at the end is the oldest state of the line. Programs never access a VA Pac line directly in this file, but first obtain the number of the top of the chain by consulting the Index file (AN).

.THE INDEX FILE (AN)

```
     .Organization : VSAM-KSDS
     .Recsize      : 54
     .C.I. size    : 4,096 for index and data
     .Key          : 43 (position 0)
     .Utilization  : Batch and On-line
     .Dsname       : &INDUV..&ROOT.&FILE.AN
     .Size         : 75  records per C.I. of 4,096
```

The Index file, by the contents of its key, describes the various views of the VA Pac Database that are offered to the user. It identifies a VA Pac line according to its position in the database and from the point of view of the consultation. It also ensures all the additional technical functions.
The essential information provided here is the internal number of the VA Pac line to which the index points.

.THE GENERATION-PRINTING REQUEST FILE (AG)

```
     .Organization : VSAM-KSDS
     .Recsize      : 150
     .C.I. size    : 4,096
     .Key          : 27 (position 0)
     .Utilization  : Batch and On-line
     .Dsname       : &INDUV..&ROOT.&FILE.AG
     .Size         : 27  records per C.I. of 4,096
```

Storage area for users which allows input and manipulation of user generation-printing requests. This file is rather small, however it is subject to considerable update activities on a daily basis. It is saved by the SVAG procedure.

It is initialized, restored and can be reorganized by the REAG procedure.

.THE JOURNAL FILE (AJ)

```
     .Organization : VSAM-RRDS
     .Recsize      : 167
     .C.I. size    : 512
     .Utilization  : Batch and on-line
     .Dsname       : &INDUV..&ROOT.&FILE.AJ
     .Size         : 3 records per C.I. of 512
```

All modification transactions that pass through the database whether in batch or on-line, are saved for two reasons. First, to allow database restoration should the standard security system ever fail. Second, this information may be used for statistical purposes.

These transactions are generally stored in the Journal Backup (PJ) file. The Journal file is only used as a transition between the time the transactions are processed and the time the ARCH procedure sends them to their final destination: PJ file.

NOTE:

The transactions contained in the Generation-Printing Request file (AG) are not saved in the Journal file (AJ).

The database backup is made of three sequential generation files.

.BACKUP FILE OF THE DATABASE (PC)

```
    .Organization : sequential, generation
    .DSNAME       : &INDUN..&ROOT.&FILE.PC(n)
    .DCB          : RECFM=VB,LRECL=155,BLKSIZE=27,280
    .Utilization  : Batch
    .Size         : 149 bytes per data
                    + 55 bytes per index
```

This is a common backup file of the VA Pac components: Index (AN), Data (AR), in a sequential format.

.BACKUP FILE OF THE JOURNAL (PJ)

```
    .Organization : sequential, generation
    .DSNAME       : &INDUN..&ROOT.&FILE.PJ(n)
    .DCB          : RECFM=FB,LRECL=167,BLKSIZE=6,179
    .Utilization  : Batch
```

The purpose of this file is to store all update transactions that have affected the VA Pac Database since the installation and that have passed through the Journal file (AJ). When its size becomes incompatible with operation requirements, the ARCH procedure divides PJ into several files and only the most recent one is used on a daily basis.

.BACKUP OF THE GENERATION-PRINTING REQUEST FILE (PG)

```
    .Organization : sequential, generation
    .DSNAME       : &INDUN..&ROOT.&FILE.PG(n)
    .DCB          : RECFM=FB,LRECL=150,BLKSIZE=6,150
    .Utilization  : Batch
```

The purpose of this file is to backup the generation-print requests and to reorganize them using the REAG procedure.

2.9.2. SEQUENTIAL BACKUP FILES

SEQUENTIAL BACKUP FILES

The database backup is made of three sequential generation files.

.BACKUP OF THE DATABASE (PC)

```
    .Organization : sequential, generation
    .DSNAME       : &INDUN..&ROOT.&FILE.PC(n)
    .DCB          : RECFM=VB;LRECL=155;BLKSIZE=27,280
    .Utilization  : Batch
    .Size         : 149 bytes per data
                    + 55 bytes per index
```

This is a common backup file of the VA Pac components: Index (AN), Data (AR), in a sequential format.

.BACKUP OF THE JOURNAL (PJ)

```
    .Organization : sequential, generation
    .DSNAME       : &INDUN..&ROOT.&FILE.PJ(n)
    .DCB          : RECFM=FB;LRECL=167;BLKSIZE=6,179
    .Utilization  : Batch
```

The purpose of this file is to store all update transactions that have affected the VA Pac Database since the installation, and that have passed through the Journal file (AJ). When its size becomes incompatible with operation requirements, the ARCH procedure divides PJ into several files and only the most recent one is used on a daily basis.

.BACKUP OF THE GENERATION-PRINTING REQUEST FILE (PG)

```
    .Organization : sequential, generation
    .DSNAME       : &INDUN..&ROOT.&FILE.PG(n)
    .DCB          : RECFM=FB;LRECL=150;BLKSIZE=6,150
    .Utilization  : Batch
```

The purpose of this file is to backup the generation-print requests and to reorganize them using the REAG procedure.

2.9.3. PEI FILES

'PEI' FUNCTION (PRODUCTION ENVIRONMENT INTERFACE) FILES

Three additional evolving files are managed by the system either on-line or in batch mode when the PEI function is available (Refer to the chapter corresponding to this function).

These files contain all data necessary for the management of the production environment.

.BATCH PRODUCTION ENVIRONMENT FILE (AB)

```
.Organization: VSAM-KSDS
.Recsize     : 110
.CI Size     : 1,024 (index) 4,096 (data)
.Key         : 26 (position 0)
.Utilization : batch and on-line(consultation only)
.Dsname      : &INDUV..&ROOT.&FILE.AB
.Size        : 37 records per C.I. of 4,096
```

.ON-LINE PRODUCTION ENVIRONMENT FILE (AC)

```
.Organization : VSAM-KSDS
.Recsize      : 110
.CI Size      : 1,024 (index) 4,096 (data)
.Key          : 26 (position 0)
.Utilization  : batch and on-line
.Dsname       : &INDUV..&ROOT.&FILE.AC
.Size         : 37 records per C.I. of 4,096
```

.PEI BACKUP FILE (PP)

```
.Organization : sequential with generation
.DSNAME       : &INDUN..&ROOT.&FILE.PP(n)
.DCB          : RECFM=FB;LRECL=110;BLKSIZE=6,160
.Utilization  : batch
```

2.9.4. DSMS FILES

DEVELOPMENT AND SUPPORT MANAGEMENT SYSTEM (DSMS) FILE

An additional evolving file is accessed on-line or in batch mode when the DSMS function is available on-site (for more details, refer to the DSMS Operations Manual).

This file contains the list of VA Pac entities which are to be updated for each CHANGE NUMBER (the Change Number is entered on the VisualAge Pacbase sign-on screen).

.DSMS PACBASE ENTITY FILE (DC)

```
.Organization : VSAM-KSDS
.Recsize      : min: 50, max: 168
.CI Size      : 4,096
.Key          : 31 (position 2)
.Utilization  : accessed by batch and on-line
                updates.
.Batch DDNAME : PAC7DC (UPDT, REST and RESY procs.)
.CICS DDNAME  : xxy0DC
 (xx = ROOTD, y = ROOT2 of the DSMS installation)
```

This file is allocated and initialized at the time of the installation of the DSMS Function. The definition supplied when installing VA Pac must be used when the DSM variant has been required, and when the DSMS function has not yet been installed on the site.

2.9.5. PAF FILES

PACBASE ACCESS FACILITY (PAF) FILES

.On-line PAF work file (PA) for PAF-TP and PUF-TP

Indexed work file enabling the PAF and PUF functions to be used for all on-line user programs accessing databases which have the same root.

```
.Organization : VSAM-KSDS
.Recsize      : average: 200, max: 539
.CI Size      : 4,096
.Key          : 37 (position 2)
.Utilization  : updated by the xxTPST and/or xxTPWS
                sub-programs called by On-Line user
                programs.
.Dsname       : &INDSV..&ROOT.&ROOT.PA
.DDNAME       : xxxxPA
                (xx = ROOT)
```

.Batch PAF work file

Indexed work file enabling the PAF function to be used for all user batch programs accessing databases which have the same root. This file is allocated for the job duration and is destroyed at the end of the job.

```
.Organization : VSAM-KSDS
.Recsize      : average: 170, max: 468
.CI Size      : 4,096
.Key          : 12 (position 0)
.Utilization  : updated by the PBBT98 sub-program,
                which is called by PBBTST/PBBTWS
                (PAF sub-programs called by the
                 user programs).
.Dsname       : &INDUV..SYSPAF.&USER
.DDNAME       : SYSPAF
```

PAF-PDM EXTENSION

.Extraction path file (GS), containing the user's extractors and macro-commands:

```
.Organization : VSAM-KSDS
.Recsize      : 203
.Key          : 25 (position 0)
.Utilization  : Batch and on-line
.Dsname       : &INDUV..&ROOT.&ROOT.GS
```

2.9.6. Pac/Impact FILES

Pac/Impact FILES

. File of already-impacted criteria (FQ)

```
.Organization:  Sequential with generation
.DCB          :  RECFM=FB;LRECL=100;BLKSIZE=21600
.DSNAME       :  &INDUN..&USER.&ROOT.&FILE.FQ(n)
.Utilization  :  Memorize those impact search criteria
                 that have already been processed
```

. Search criteria or entry point file (FH)

```
.Organization:  Sequential with generation
.DCB          :  RECFM=FB,LRECL=160,BLKSIZE=24000
.DSNAME       :  &INDUN..&USER.&ROOT.&FILE.FH(n)
.Utilization  :  Memorize impact search criteria for the
                 next IANA execution
```

. Reduced file of criteria for purge (FR)

```
.Organization:  Sequential with generation
.DCB          :  RECFM=FB,LRECL=72,BLKSIZE=21600
.DSNAME       :  &INDUN..&USER.&ROOT.&FILE.FR(n)
.Utilization  :  Purge the impact search criteria in a
                 text editor
```

. Impact result file (FO)

```
.Organization:  Sequential with generation
.DCB          :  RECFM=FB,LRECL=260,BLKSIZE=26000
.DSNAME       :  &INDUN..&USER.&ROOT.&FILE.FO(n)
.Utilization  :  Memorize all the results of the impact
                 analysis.
```

. File of entities to be analyzed (FP)

```
.Organization: VSAM-KSDS
.Recsize      :  9
.CI size      :  4096
.Key          :  9 (position 0)
.DSN          :  &INDUV..&USER.&ROOT.&FILE.FP
.Utilization  :  Restrict the impact analysis to those
                 entities specified in the file
```

2.9.7. PAC/TRANSFER FILES

PAC/TRANSFER FILES

Parameter file (UV):

This file is used to control the various Pac/Transfer processes.

The creation or update of this file --via the TRUP procedure-- stores the Transaction Sets, which define the various transfer processes envisionned for the site. (Each SET corresponds to a specific parameterization.)

The processes of the Pac/Transfer facility can thus process a unique SET, a list of SETS, or all the SETS, depending on current requirements.

A number of checks against the VisualAge Pacbase database are performed by the TRUP procedure.

For further details, see the description of the TRUP procedure in the Batch Procedures: Administrator's Guide.

2.9.8. VA PAC-VA SMALLTALK INTERFACE FILES

'VISUALAGE SMALLTALK - VISUALAGE PACBASE INTERFACE' FILES

Character-correspondence table

This table lists all the characters used in VisualAge Smalltalk identifiers that are not valid for VisualAge Pacbase codes, as well as their replacement characters.

It is shipped as an empty file, which should be filled by the user as described in the description of the VUP1 procedure, in the Batch Procedures: User's Guide.

```
     .Organization:  Sequential
     .Recsize     :  80
```

2.9.9. VA PAC-TEAMCONNECTION INTERFACE FILES

'VISUALAGE PACBASE - TEAMCONNECTION BRIDGE' FILES

. Target-library and -session file (TS)

```
.Organization:  VSAM-KSDS
.Recsize     :  80
.CI size     :  1024
.Key         :  14 (position 1)
.DSN         :  $INDSV..$ROOT.$ROOT.TS
.Utilization :  batch
```

2.10. COMPLEMENTARY LIBRARIES AND FILES

COMPLEMENTARY LIBRARIES AND FILES

THE DICTIONARY EXTENSION LIBRARY (SYDI)

```
. Size         : approximately 100 blocks of 6,080 bytes
. Organization: PDS
. DCB          : Recfm=FB;Lrecl=80;Blksize=6,080
. Dsname       : &INDSN..&ROOT.&ROOT.SYDI
```

. PAF FUNCTION: in member PAFDIC, batch transactions on Data Element, Data Structure and Segment entities required by PAF Table descriptions and planned for integration in a Dictionary.

. PAF FUNCTION EXTENSION: in member PAFPTEX, batch transactions on Data Element and User Entities .PPTEX "Extraction Master Path", planned for integration in a Dictionary.

. In member PAFJCL, a JCL, sample of a batch user program using PAF.

. In member PTEXJCL, a sample of a user extractor execution JCL (PAF extension).

. PQC function: in member PQCRULE, the standard quality rule file.

. PQC FUNCTION PERSONALIZATION: in member PQCUPDT, batch transaction on Data Element and .QPAQC User Entities planned for integration in a Dictionary.

. ERROR MESSAGE UPDATE UTILITIES (user applications): in member EMUPDT, batch transactions on Segment Entities and Batch Programs (UTEMLD and UTEMUP), planned for integration in a Dictionary. These will be used for creating error-message loading and update programs for a given application, according to a site's requirements.

Member	Contents or format	Comments
Specific members of the PAF and PAF+ functions		
PAFDIC	Batch transactions	PAF Dictionary
PAFJCL	JCL	PAF batch user program sample
PAFPTEX	Batch transactions	PAF+ Dictionary
PTEXJCL	JCL	User extractor execution JCL sample
Specific members of the Pacbench Quality Control function		
PQCRULE	Sequential file	Standard rules
PQCUPDT	Batch transactions	PQC Dictionary
Error message update utility source (user applications)		
EMUPDT	Batch transactions	Programs and Segments
Specific members of the Documented COBOL Option (DCOB)		
DCOBDEF	DEFINE work file	DSN parametered by PRMSYS
DCOBDEL	DELETE - -	- - - -
DCOBIN	Description param. COBOL SYSPRINT	(See chapter dedicated to the DCOB procedure.)
DCOBOUT	IEBGENER SYSIN	To justify the COBOL SYSPRINT on 133 characters
DCOBSRT	SORT FIELDS	DO NOT MODIFY

THE VINS-PROCEDURE INPUT LIBRARY: SYEN

```
. Size         :  approximately 25 tracks
. Organization:  PDS
. Lrecl        :  117
. Blksize      :  7,020
. Utilization  :  Batch
. Dsname       :  &INDSN..&ROOT.&ROOT.SYEN
```

This file contains the input to the VINS procedure. (For further details, refer to the corresponding chapter in the Batch Procedures, Administrator's Guide.)

The library contains:

VGEDIC : Definitions of entities in the VisualAge Smalltalk Dictionary that are to be integrated in the VisualAge Pacbase Dictionary.

This integration is required for correct operation of the VisualAge Pacbase-VisualAge Smalltalk Bridge.

TEAMDIC: Definitions of entities in the 'TeamConnection' Dictionary.

This integration is required for correct operation of the VisualAge Pacbase - TeamConnection Bridge.

THE PACDESIGN METHODOLOGY LIBRARY: SYMT

```
. Size        : approximately 100 blocks of 6,080 bytes
. Organization: PDS
. DCB         : Recfm=FB;Lrecl=80;Blksize=6080
. Dsname      : &INDSN..&ROOT.&ROOT.SYMT
```

. For the WorkStation to operate, two types of integrations in VA Pac are required:

- In the Dictionary: integration of transactions associated with the Data Elements and User entities of the Pacdesign methodology in use.

- In the VA Pac system: integration of transactions defining methodology choices (User Parameters).

. These transactions are grouped in the following files:

Member	Contents	Proc.
DESMER	Batch transactions MERISE methodology	UPDT
PARMMER	Parameterization MERISE Methodology	PARM
DESADM	Batch transactions SSADM Methodology	UPDT
PARMADM	Parameterization SSADM Methodology	PARM
DESYSM	Batch transactions YSM Methodology	UPDT
PARMYSM	Parameterization YSM Methodology	PARM
DESIFW	Batch transactions IFW methodology	UPDT
DESIFWP	Pre-loading IFW methodology	UPDT
PARMIFW	Parameter setting IFW methodology	PARM
DESOMT	Batch transactions OMT methodology	UPDT
PARMOMT	Paramétrage OMT methodology	PARM

DO NOT MODIFY THE CONTENTS OF THESE FILES!

THE MULTI-SCREEN OLSD VARIANT SOURCE LIBRARY

This additional library needs to be loaded only for users of the multi-screen variant of the OLSD function.

It contains the source sub-programs for the management of the screen message "ZAR980" for all dedicated generators for which this variant is available.

```
     . Size        : approximately 250 blocks of 6,080 bytes
     . Organization: PDS
     . DCB         : Recfm=FB;Lrecl=80;Blksize=6080
     . Dsname      : &INDSN..&ROOT.&ROOT.MSO
```

Member	Contents
ZARCVS	MVS/CICS and VSE/CICS Cobol II
ZARG7	GCOS7/TDS
ZARG8	GCOS8/DMIV and TP8
ZARICL	ICL
ZARBUR	Unisys A
ZARDEC	DEC (characters)
ZARDE2	DEC (fields)
ZARTRM	DEC (assembler)
SCRDEC	DEC sub-program
PACVMSS	Digital VMS
VMSUTIL	Digital VMS
HPFORM	HP3000 screen-message processing
ZARMF1	Microfocus sub-program
SCRCODIF	Microfocus sub-program
SCRIOPAR	Microfocus sub-program
SCRPEINT	Microfocus sub-program
SCRSAISI	Microfocus sub-program
ZARMFO	Product-reserved
SCRMFO	Product-reserved
WEBCVS	MVS/CICS (WEB)

THE LIBRARY OF SQL SOURCES FOR ACCESS TO DB2 CATALOG

This additional library needs be loaded by users of the SQL variant (ex-DB2 variant) only.

```
. Size        : approximately 25 blocks of 6,080 bytes
. Organization: PDS
. DCB         : Recfm=FB;Lrecl=80;Blksize=6080
. Dsname      : &INDSN..&ROOT.&ROOT.SQL
```

SQL INSTALLATION VARIANT

Member	Contents
PACR12	Update of DB2 catalog DB2 program COBOL source to be compiled under name xxRS12 (xx=ROOT).

INTERACTIVE SQL APPLICATION IN VA PAC

With the B.....GEN choice, the xxQL45 program calls the xxQL46 program, which generates the update SQL and calls the SQL update module xxRS12.

The xxQL45 and xxQL46 programs are VA Pac standard programs shipped in the form of executable modules.

The xxRS12 module, shipped in the form of a COBOL source program in member PACR12, must be prepared and installed as any DB2 standard program:

- Pre-processor preparation, resulting in a COBOL source and a DBRM (same name as the COBOL source).
- Compilation and link-edit of the COBOL source, resulting in an executable program (xxRS12).

The plan is built with the DBRM, using the DB21 TSO application (this being the BIND operation).

The CICS RCT table is updated with the VA Pac transaction code and the code of the previously built plan.

SPECIAL AUTHORIZATIONS

The TSO UserID used for the BIND must allow for the consultation of the following catalog tables:

- SYSIBM.SYSDATABASE
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS

NOTES:

. Updates are performed via SQL dynamic orders.

. DB2 tables or views which are accessed or updated are not qualified, i.e. the user's CICS UserID implicitly qualifies the DB2 object's name.

THE LIBRARY OF COBOL/VS SUB-PROGRAMS: MCOBVS

This library must be loaded for the development of user PAF programs in COBOL/VS. The standard-supplied PAF sub-programs are in COBOL II.

The PAF COBOL/VS sub-programs, as well as those of the Security System Interface (where this is operating on the site), must be loaded in the load-module libraries of the PAF user programs.

- Size : approximately 100 blocks of 6,144 bytes
- Organization: PDS
- DCB : Recfm=U,Blksize=6144
- Dsname : user-defined

CODE	FUNCTION	Comments
		BATCH
PBBTST	PAF	Sub-program called by user batch pgms.
PBBTWS	-	-
PBBT98	-	-
PACSECB	SEC	Security System Interface
		CICS (xx=ROOT, on-line program prefix)
xxTPST	PAF	Sub-pgm called by on-line user pgm.
xxTPWS	-	-
xxSECT	SEC	Security System Interface

3. ENVIRONMENT

3.1. INTRODUCTION

INTRODUCTION

This chapter details the environment and the resources required by VisualAge Pacbase, so as to help you to determine the necessary disk space.

3.2. ON-LINE ENVIRONMENT (CICS)

ON-LINE ENVIRONMENT (CICS)

The monitor in use is CICS ESA (release 3.1 +).

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

It must include the option SPOOL=YES, for the submission of batch jobs (GP choice).

The Data (AR), Index (AN), Journal (AJ), Generation-Printing Requests (AG) as well as the Error Messages (AE) and User Parameter (AP) files are updated on-line and must be protected by the 'DYNAMIC BACKOUT' option.

The same applies for the on-line production environment file (AC) of the PEI function.

The maximum number of strings to be used per file is two. This number can thus be adjusted without restriction according to the available space in the CICS partition. (See Subchapter 'Efficiency Enhancement.)

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

The size of the COMMAREA in use is 3.5 K.

The average size of an on-line program is 48K, the largest being 115K.

GENERAL INFORMATION - HOW THE SYSTEM RUNS

The general characteristics are:

VISUALAGE PACBASE TRANSACTION:

A single transaction code is used. It points to a 'monitor' program whose purpose is to chain together the different programs that make up the system. No screen is directly driven by this program. (Program code: R000).

. Each screen that can be used for updating is associated with a given program (for example: QC00 updates the definition of a data element).

. Screens that display the 'simple' lists are processed in a single program (QH01).

. The so-called 'special' lists, such as the cross-references, keywords, etc., are processed in specific programs (QS02, QS03, QS04 and QS05).

. Menus are processed in QS06.

. Some programs are called by the LINK command. This is the case for the program that processes the CHOICE field (R100), as well as for the program that validates the data element formats (R200).

. In case of a system ABEND, an 'ABEND MAP' is generated. The display program for this map (R500) is called by XCTL.

. Updates are serialized; in other words, the system protects the database against concurrent accesses by placing all update transactions in a wait queue (ENQUEUE and DEQUEUE in the update cycle).

A specific program is called when the user quits the VA Pac transaction (program code: R005). This program displays the message:

CURRENT PACBASE CONVERSATION IS SAVED

when users leave the transaction using the CHOICE .12 or the corresponding PFkey (standard : PF12), and executes an "EXEC CICS RETURN".

In order to avoid giving control to CICS, the R005 program may be replaced by a user-program which will, for instance, call a user transaction.

VA Pac sends a 3-character COMMAREA at the end of the transaction:

position 1-2: Blank or 12 (conversation back-up)
position 3 : 'A' or 'F': language code (install. parameter)

The program replacing R005 may use the COMMAREA, and display the message 'CURRENT CONVERSATION SAVED' when the first two positions are '12' and the third is 'A'.

USER PARAMETER AND PEI TRANSACTION:

A specific transaction code is used, independently from the database management, in order to:

. Manage user parameters (standard),
. Manage the production environment (available with the PEI function).

UPPERCASE AND LOWERCASE MANAGEMENT

The VisualAge Pacbase system has its own input character management system:

. All codes entered in lowercase are automatically transformed into uppercase,

. All entity names, as well as text, remain in lowercase. Implicit keywords drawn from entity names are changed into uppercase words.

In order to inhibit the lowercase to uppercase switch, the letter 'X' should be entered in the ACTION CODE field.

In CICS ESA Release 3.2 or higher, 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 ESA 3.1, and from CICS ESA 3.3, 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.
At the time of installation, 3 routine versions are offered. The xxUCTR routine is to be used in a CICS ESA 3.1 context, the xxUCTZ routine (renamed xxUCTR) is to be used with a CICS ESA release higher or equal to 3.3, and the xxUCTX routine (renamed xxUCTR) is to be used with a CICS ESA release 3.2.
The latter can be used to deactivate the modification of the UCTRAN parameter in all CICS releases, which is useful if you wish to work always in uppercase letters.

In Batch mode, the character transformation follows the rules that apply to On-Line mode.

If you wish to work in uppercase mode only, use the functionalities of the editor that contain the PACBASE batch transactions, to ensure automatic switch from lowercase to uppercase, since PACBASE itself does not perform this transformation in batch mode.

3.3. ACCESS METHODS (VSAM)

ACCESS METHODS (VSAM)

The VA Pac system 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). This share option is the only protection of the PACBASE system against simultaneous updates, both in on-line and batch modes.

Production Environment Interface:

As the AB file may be updated in batch mode by simultaneous GPRT procedures, it is not protected (SHARE OPTION 4).

All batch procedures include DELETE/DEFINE steps to take care of file reloading. Therefore, the files do not need to include the REUSE default option.

3.4. BATCH ENVIRONMENT

THE 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.

The JCL provided for installation allows for extensive blocking factors for the management procedures which process large volumes of data, in order to improve performance. The blocking factor chosen in these procedures (REOR, MLIB, GPRx) is approximately half a track on a 3,390 disk. Therefore, the size of the virtual memory can increase to 4,096 K (as in the case of the PTU220 program in the REOR procedure).

Using BATCH LSR in the UPDT and GPRT procedures also influences the size of the processing region (see Subchapter'Efficiency Enhancement' below.)

If these blocking factors are not suitable (other types of disk, reduction of the virtual memory size), they may be modified.

3.5. FILE SIZE

FILE SIZE

The total amount of space needed for the files depends on the size of the applications managed by the system.

The following table indicates the disk space necessary for a standard installation:

VisualAge Pacbase installation size	Approximation in millions of bytes		
	TOTAL	VSAM	Non-VSAM
Installation total	55	24	31
System total	35	7	28
Installation test user file total	20	17	3

The size of the VA Pac system files remains fixed.

The size of the user files is explained on the next page. It is possible to get a rough estimate of the space to be allocated by allowing 400 bytes per VA Pac data, all files included.

The total amount of space needed for the EVOLVING (USER) FILES can be calculated by assuming the following:

Let NPAC be the number of VA Pac records, all libraries and sessions included. Then the following applies:

(AR) Data File: NPAC records of 140 bytes.

(AN) Index File: Approx. 3 * NPAC records of 54 bytes (on average, an element is used 3 times), plus the free space (FSPC) added when defining the file.

(AG) Generation-Printing Request File: Generally takes up little space. As an estimate, it should be able to contain approximately a hundred or so requests per user accessing the system (150 bytes).

(AJ) Journal File: It must contain enough space for all update transactions, in batch or on-line mode, that may pass between 2 re-initializations of the Journal file. A VA Pac transaction corresponds to one record of the Journal file (167 bytes).

(AP) User Parameter File: Takes up minimal space. It contains the site JCL lines and optional modifications of standard error messages (80 bytes per record).

CALCULATION OF VSAM FILE SIZE

```
KSDS file:
No. of Rec./CI = (CISIZE DATA - 10 - Free space CI) / RECSZ

RRDS file:
No. of Rec./CI = (CISIZE - 4) / (RECSZ + 3)
```

Example

```
For a database containing 1,000,000 data records, there are:

AN file: DATA CISIZE(4096) RECSZ(54) FSPC(10,6)

   .CI free space = 10 % of 4096 = 409
   .No. of records per CI = (4096 - 10 - 409) / 54 = 71
   .No. of CI's per 3380 disk track = 10
    (See 'VSAM Administration Guide' IBM GC26-4066-1)
   .Maximum No. of CI's/CA (1 CA = one 3380 cylinder) =
                                                10 * 15 = 150
   .CA free space = 6 % of 150 = 9
   .No. of CI's used for each CA = 150 - 9 = 141
```

Thus, for 3,000,000 data records (3 * NPAC), there are about:

```
      3,000,000 / 71 = 42,254 CI's,
        42,254 / 141 = 300 DATA CA's,
```

which represents about 300 cylinders of a 3380 disk.

AR file: CISIZE(2048) RECSZ(140)

```
   .No. of records per CI = (2048 - 4) / (140 + 3) = 14
   .No. of CI's per 3380 disk track = 18
    (See 'VSAM Administration Guide' IBM GC26-4066-1)
   .No. of CI's/CA (1 CA = one 3380 cylinder) =
                                                18 * 15 = 270

Thus, for 1,000,000 data records (NPAC), there are about:

      1,000,000 / 14 = 71,429 CI's,
        71,429 / 270 = 265 CA's,

which represents about 265 cylinders of a 3380 disk.
```

The total represents about 400 million bytes for the two most important evolving files of the database.

PRODUCTION ENVIRONMENT INTERFACE OPTION (PEI)

The AB and AC files contain the same data. Therefore they should have the same amount of space, calculated as follows:

.record length: 110
.number of load-module libraries: NLM
.number of entities in production: NEP
.number of generated entities per load-module library: NEG

The number of storage bytes needed should be:

110 * (NLM + (2 * NEP) + NEG).

3.6. EFFICIENCY ENHANCEMENT

EFFICIENCY ENHANCEMENT

The performance of the VisualAge Pacbase system may be improved in batch or on-line mode in several ways, depending on the environment and on the size of the managed VA Pac databases.

This chapter provides information necessary for efficiency enhancement.

The file, program and JCL installation default options may be modified according to the suggestions described below. It is also possible to enhance efficiency by modifying the system environment, which is not described in this chapter since it depends on the specifications of each site.

Moreover, it is recommended not to modify too many chosen default options, so that the VA Pac Database Manager can retrieve these modifications easily when reinstalling the system.

3.6.1. INSTALLATION OF THE INDEX AND DATA FILES

INSTALLATION OF THE INDEX (AN) AND DATA (AR) FILES

Where possible, the AN and AR files should be copied onto separate volumes.

For large VA Pac Databases (more than one million data items), it may also be appropriate to allocate the AN and AR files onto several volumes.

Since the AN file is used in its entirety by the system, it is possible to carry out a multi-volume installation simply by modifying the file's DEFINE in the DFxxnnAN member of the VA Pac Database parameter PDS.

Moreover, about 80 percent of the VA Pac activity use the data stored at the beginning of the AR file, in the case of normal operations carried out by a development team. Therefore, the allocation of the AR file on several volumes does not in itself provide access optimization.

In order to take advantage of this multi-volume distribution, a standard procedure, called Storage Optimization of Multi-Volume Data (STOP) is provided. It allows for a wide distribution of the most frequently used data within the whole file.

The STOP procedure, combined with the DEFINE of the AR file (DFxxnnAR member of SY PDS) on several volumes, must therefore be used before restoration of the database.

(For a description of the STOP procedure and details concerning data distribution parameterization, see the Batch Procedures: Administrator's Guide.)

3.6.2. VISUALAGE PACBASE STRUCTURE UNDER CICS

VISUALAGE PACBASE STRUCTURE UNDER CICS

As a general rule, each VA Pac line type is processed by a specific program in the PPT. There are two types of programs:

- Programs that can update the database,

- Programs that can only read the database.

There may be several programs which correspond to one VA Pac transaction:

The first loaded program, called a 'monitor', sends control through XCTL to the program able to process the type of line displayed on the screen.

This program executes the following instructions:

. Screen read (RECEIVE).

. If an update is detected:

- Enqueue execution on a character string specific to the update database,

- Read update of the first record on the AR file,

- Rewrite of this record,

- Loop on requested updates, which may contain per updated line:

. Several positionings or sequential reads on the AN file (STARTBR, RESETBR, or READNEXT),
. Several simple reads on the AR file (READ),
. One or two write commands on the AR file (WRITE or READ UPDATE, REWRITE),
. Several write commands on the AN file (WRITE or READ UPDATE, REWRITE),
. A write command on the AJ file (WRITE),

- READ UPDATE of first record in the AR file,

- REWRITE of this record,

- DEQUEUE execution corresponding to the ENQUEUE.

. If the CHOICE field is entered: LINK to the CHOICE field decoding program.

. If the CHOICE is valid: XCTL on the corresponding program.

. Display step broken down as follows:

- Several positionings or sequential reads on the AN file (STARTBR, RESETBR, or READNEXT),

- Several simple reads on the AR file (READ).

. SEND (send screen).

. RETURN TRANSID (end of transaction).

NOTES

ENQUEUE/DEQUEUE operates in such a way that updates are serialized.

The program structure is such that, at a given time, only one BROWSE or READ UPDATE is active on the AN file; only one string is used by a transaction during a time lapse greater than one input/output.

3.6.3. RESOURCES REQUIRED UNDER CICS

RESOURCES REQUIRED UNDER CICS

In order to improve the VA Pac system efficiency under CICS, two objectives should be pursued:

- Reduce program loading time,
- Make VSAM buffer management more efficient.

1. Program management

The first solution is fairly easy to implement: it requires a frequency survey on program runs. As a result, the programs that are most often used are said to be resident in the CICS partition (RESIDENT(YES) in CSD).

Example

Here are the programs most commonly used by the VA Pac system:

```
xxR000                Monitor

xxR100                CHOICE field decoding

xxR200                Format validation

xxQ300  xxN300        Update of segment -CE lines

xxQT00  xxNT00        Update of text -D lines

xxQP00  xxNP00        Update of -P lines

xxQI00  xxNI00        Update of dialogue -CE lines
xxQI01  xxNI01                    -
xxQI02  xxNI02                    -

xxQH20  xxNH20        Update of dialogue -CS lines.
```

2. Buffer management

Three parameters associated with the AN and AR files must be entered in CSD: STRINGS, DATABUFFERS, and INDEXBUFFERS.

- STRINGS:

It specifies the number of STRINGS or requests which can be executed in parallel by VSAM on the same field: usually, a STRING is used during an input/output on disk.

Certain commands, however, do not follow this rule:

READ UPDATE and STARTBR or RESETBR reserve a STRING up to the REWRITE or UNLOCK and ENDBR commands, respectively.

- DATABUFFERS:

It specifies the number of DATA buffers to be allocated by VSAM when the file is opened. A minimum of two buffers is required: one for the read of control interval DATA, one for the control interval 'split'.

Default: DATABUFFERS = STRINGS + 1

- INDEXBUFFERS:

It specifies the number of INDEX buffers to be allocated by VSAM when the file is opened. The minimum number of buffers per request is one. It is preferable, however, to increase this number because for a file with three INDEX levels, three EXCP's are required to point to the searched CI DATA for each on-line request.

It is recommended to allocate:

AT LEAST: one buffer per INDEX level, plus one buffer per STRING (STRING value).

NO MORE THAN: one buffer per CI of INDEX SET, plus one buffer per STRING.

Conclusion

A sufficient number of buffers should be provided for, according to the number of STRINGS.

However, too many buffers may lead to an excessive pagination, and therefore EXCP's.

Below is a chart of optimized values for the different parameters, according to the number of STRINGS:

NUMBER OF STRINGS	AN FILE STRINGS	AN FILE INDEX BUFFERS	AN FILE DATA BUFFERS	AR FILE STRINGS	AR FILE DATA BUFFERS
1	1	4	4	1	2
5	5	8	7	5	6
10	10	13	11	10	11
20	20	23	21	20	21
30	30	33	31	30	31
40	40	43	41	40	41

The remaining question is how to evaluate the number of concurrent requests at a given time.

3.6.4. EVALUATION OF THE NUMBER OF REQUESTS UNDER CICS

EVALUATION OF THE NUMBER OF REQUESTS UNDER CICS

If the number of requests is estimated by multiplying the number of strings necessary for a transaction by the maximum number of users, the resulting number of buffers would be too large, which would thus imply excessive memory requirements.

A preferable method is the probability approach, taking into account simple hypotheses which are relevant in analyzing how resources are managed under CICS.

Request number evaluation method

Transaction execution requests are entered according to the "POISSON distribution" law and the transaction's duration is a constant. Taking into account this duration and the frequency of requests, the number 'n' is determined so that the probability of having more than n transactions at a given time is less than 1/100,000.

The average duration of a transaction at a specific site has to be calculated, as well as the average frequency of requests. The following chart aids in finding this number 'n' based on these parameters.

This number equals the number of STRINGS indicated in the preceding chart, which allows the STRINGS, DATABUFFERS and/or INDEXBUFFERS parameters for the AN and AR files.

Time is expressed in seconds, average frequencies vary from 0.1 to 2, the duration of requests therefore varies from 10 to 0.5 seconds.

Refer to the table on the following page.

DURAT. / FREQU.	0.1	0.5	0.9	1.3	1.7	2.1	2.5	2.9	3.3	3.7	4.1	4.5	4.9
0.1	2	3	4	4	5	5	5	5	6	6	6	6	6
0.2	3	4	5	5	6	6	7	7	7	8	8	8	8
0.3	3	4	5	6	7	7	8	8	9	9	9	10	10
0.4	3	5	6	7	7	8	9	9	10	10	11	11	12
0.5	3	5	6	7	8	9	9	10	11	11	12	12	13
0.6	3	5	7	8	9	9	10	11	12	12	13	14	14
0.7	3	6	7	8	9	10	11	12	13	13	14	15	15
0.8	4	6	7	9	10	11	12	13	13	14	15	16	16
0.9	4	6	8	9	10	11	12	13	14	15	16	17	17
1.0	4	7	8	10	11	12	13	14	15	16	17	18	19
1.1	4	7	9	10	11	13	14	15	16	17	18	19	20
1.2	4	7	9	10	12	13	14	15	16	18	19	20	21
1.3	4	7	9	11	12	14	15	16	17	18	19	20	22
1.4	4	7	9	11	13	14	15	17	18	19	20	21	22
1.5	4	8	10	12	13	15	16	17	19	20	21	22	23
1.6	4	8	10	12	14	15	17	18	19	21	22	23	24
1.7	5	8	10	12	14	16	17	19	20	21	23	24	25
1.8	5	8	11	13	14	16	18	19	21	22	23	25	26
1.9	5	8	11	13	15	17	18	20	21	23	24	26	27
2.0	5	9	11	13	15	17	19	20	22	24	25	26	28

4. INSTALLATION

4.1. INTRODUCTION

INTRODUCTION

The installation procedure is broken down into three main stages:

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

The installation process requires a special tape (or cartridge) and is described in this chapter.
You should be familiar with the VA Pac System's characteristics, which are fully described in this manual, in order to properly prepare an environment for the installation procedure (disk space, CICS options, codification, etc.).

Once the environment is ready, you can proceed with the installation.

PREPARATION

Unload the initial JCL from the tape, and execute it:

. Perform a backup of the installation tape,
. Copy the module for processing the complete JCL,
. Restore the complete VA Pac installation and operation JCL.

INSTALLATION

See Subchapter 'Installation Process'.

TESTS

. On-line procedure tests,
. Batch procedure tests.

4.2. INSTALLATION TAPE

THE INSTALLATION TAPE

The installation tape, or "3480" cartridge (6,250 BPI, standard labels) contains the following files:

RANK	LABEL	LRECL	BLKS	CONTENTS
1	INST.JCL	80	11440	Initial preparation JCL.
2	INST.MOD		6144	Load module MM1JCL, utility for JCL preparation.
3	PAC.JCL	80	11440	JCL skeleton for installation and operation.
4	PAC.MBR8		6144	Batch load-modules
5	PAC.MTR8		6144	On-line load-modules
6	PAC.AE0	80	11440	Error message source file and Documenta- tion.
7	PAC.SRCDIC	80	3440	Dictionary comple- ments
8	PAC.SC	3204	3204	Batch language skeleton file
9	PAC.SG	4605	4605	OLSD and DBD skeleton file.
10	PAC.SR	4605	4605	COBOL generator skeleton file.
11	PAC.SS	4605	4605	CSSD skeleton file.
12	PAC.RES1	4605	4605	Reserved to product
13	PAC.SP	4605	4605	PAF-PDM variable skeleton file.

RANK	LABEL	LRECL	BLKS	CONTENTS
14	PAC.SF	119	11900	PAF Extension fixed skeleton file.
15	PAC.INSDIC	117	7020	VINS proc. input
16	PAC.PCTEST	155 (VB)	23405	Test database backup
17	PAC.SRCMTH	80	3440	WorkStation methodologies
18	PAC.SRCSQL	80	6080	SQL source PDS
19	PAC.SOURCE	80	6080	Source PDS: .ZAR980 screen-msg .TOPSECRET interface
20	PAC.MCOBVS		6144	COBOL/VS sub-program (PAF)
21	PAC.SFCOBVS	119	11900	COBOL/VS fixed skeleton file (XPAF)

In case of a bilingual installation (English-French), the installation tape contains the following additional files:

RANK	LABEL	LRECL	BLKS	CONTENTS
				BILINGUAL:
22	PAC.AEOB	80	11440	Error messages and Documentation.
				ENGLISH:
23	PAC.AEOA	80	11440	Error messages and Documentation.
24	PAC.SCB	3204	3204	Batch language skeleton file.
25	PAC.SGB	4605	4605	OLSD/DBD skeleton file.
26	PAC.SRB	4605	4605	COBOL generator skeleton file.
27	PAC.SSB	4605	4605	CSSD skeleton file.
28	PAC.PCB	155	23405	Test database.

For correction tapes or cartridges:

RANK	LABEL	LRECL	BLKS	CONTENTS
29	PAC.PATCH	80	6080	Patch JCL PDS
30	PAC.README	133 (FBA)	5320	"README" PDS Info. on correction release

4.3. INSTALLATION PREPARATION

INSTALLATION PREPARATION

It is recommended to copy all the VisualAge Pacbase preparation, installation, and operation JCLs in a specific PDS.

This PDS allocation is not supplied and must be performed in advance by the person responsible for the installation.

This PDS must have the following characteristics:

- Lrecl=80
- Size: about 60 tracks of a 3380 disk, 30-directory blocks.

Then, copy the initial JCL from the VA Pac installation tape ('INST.JCL') into the allocated PDS, using one of the site's utilities (IEBGENER, for example).

INITIAL JCL 1

The initial JCL contains the JOB's necessary for the generation of VA Pac complete installation and operation JCL.

This complete JCL is obtained via a parameterized skeleton JCL ('PAC.JCL'), thus allowing each site's specific needs to be taken into account.

This JCL is processed by an 'INST.MOD' file utility, which generates a JCL taking into account the parameters specified by the user.

The INITIAL JCL includes three JOBs that should be adapted to the site's requirements:

. Copy of the installation tape or cartridge provided with thr product onto a user media. This will form the VA Pac backup, and it must be used for the actual installation.

. Downloading, via IEBCOPY, of the MM1JCL utility contained in the INST.MOD file into a load-module library already existing on the site or allocated for this purpose. This operation is executed for the first installation of the system only.

. Execution of MM1JCL which creates the actual in1stallation and operation JCL.

Complete the JCL lines for this execution as follows: //STEPLIB DD DSN= <--- library containing the MM1JCL
//SYSUT1=<--- tape or cartridge number

//SYSUT2 DD DSN= <--- recipient file for the complete installation and operations JCL.
This file may be either a member of the PDS initially designed to store all the JCLs, or a sequential file of your choice.

Enter the required parameters (refer to the following subchapters).

```
THE MM1JCL OUTPUT SHOULD BE KEPT: IT MAY BE USED FOR RE-
INSTALLATIONS.
```

4.3.1. INITIAL JCL

```
$CO******** VA PACBASE 2.5 MVS/CICS $VV $DATE *****************
//PACBASE0 JOB (---),'3480',CLASS=D,MSGCLASS=A
//ALLOC   EXEC PGM=IEHINITT
//TCGI     DD  DISP=SHR,UNIT=(3480,,DEFER),VOL=(,RETAIN,SER=$CGICAR)
//TINST    DD  DISP=SHR,UNIT=(3480,,DEFER),VOL=(,RETAIN,SER=------)
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
TINST INITT SER=------,OWNER='------',DISP=REWIND
/*
//PACCOP  PROC INDEX='$INPRO',NAME=XXX,LAB=N
//GENER   EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  DUMMY
//SYSUT1   DD  DSN=&INDEX..&NAME,DISP=SHR,
//         VOL=(,RETAIN,REF=*.ALLOC.TCGI),LABEL=&LAB
//SYSUT2   DD  DSN=&INDEX..&NAME,DISP=(,KEEP),
//         VOL=(,RETAIN,REF=*.ALLOC.TINST),LABEL=&LAB,
//         DCB=*.SYSUT1
//         PEND
//STEP1    EXEC PACCOP,LAB=01,NAME=JCL,INDEX=INST
//STEP2    EXEC PACCOP,LAB=02,NAME=MOD,INDEX=INST
//STEP3    EXEC PACCOP,LAB=03,NAME=JCL
//STEP4    EXEC PACCOP,LAB=04,NAME=MBR8
//STEP5    EXEC PACCOP,LAB=05,NAME=MTR8
//STEP6    EXEC PACCOP,LAB=06,NAME=AE0
//STEP7    EXEC PACCOP,LAB=07,NAME=SRCDIC
//STEP8    EXEC PACCOP,LAB=08,NAME=SC
//STEP9    EXEC PACCOP,LAB=09,NAME=SG
//STEP10   EXEC PACCOP,LAB=10,NAME=SR
//STEP11   EXEC PACCOP,LAB=11,NAME=SS
//STEP12   EXEC PACCOP,LAB=12,NAME=RES1
//STEP13   EXEC PACCOP,LAB=13,NAME=SP
//STEP14   EXEC PACCOP,LAB=14,NAME=SF
//STEP15   EXEC PACCOP,LAB=15,NAME=INSDIC
//STEP16   EXEC PACCOP,LAB=16,NAME=PCTEST
//STEP17   EXEC PACCOP,LAB=17,NAME=SRCMTH
//STEP18   EXEC PACCOP,LAB=18,NAME=SRCSQL
//STEP19   EXEC PACCOP,LAB=19,NAME=SOURCE
//STEP20   EXEC PACCOP,LAB=20,NAME=MCOBVS
//STEP21   EXEC PACCOP,LAB=21,NAME=SFCOBVS
//STEP22   EXEC PACCOP,LAB=22,NAME=AE0B
//STEP23   EXEC PACCOP,LAB=23,NAME=AE0A
//STEP24   EXEC PACCOP,LAB=24,NAME=SCB
//STEP25   EXEC PACCOP,LAB=25,NAME=SGB
//STEP26   EXEC PACCOP,LAB=26,NAME=SRB
//STEP27   EXEC PACCOP,LAB=27,NAME=SSB
//STEP28   EXEC PACCOP,LAB=28,NAME=PCB
//STEP29   EXEC PACCOP,LAB=29,NAME=PATCH
//STEP30   EXEC PACCOP,LAB=30,NAME=README
//PACBASE1  JOB (---),'MM1JCL',CLASS=D,MSGCLASS=A
//COPY    EXEC  PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,10)
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,10)
//IM       DD DISP=OLD,UNIT=3480,VOL=(,RETAIN,SER=------),
//         DSN=INST.MOD,LABEL=2
//OM       DD DISP=SHR,DSN=---.---.---
//SYSIN    DD *
 C  I=((IM,R)),O=OM
/*
//PACBASE2  JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//MM1JCL EXEC PGM=MM1JCL
//STEPLIB  DD DISP=SHR,DSN=---.---.---
//SYSOUT   DD SYSOUT=A
//SYSUT1   DD DSN=PAC.JCL,DISP=OLD,
//         UNIT=3480,VOL=(,RETAIN,SER=------),LABEL=3
//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 DU JCL D'INSTALLATION PAR 'MM1JCL'
```

```
//*      ****************************************
//*
//*      .MODIFIER LA LISTE DES COMMANDES FOURNIES EN PRECISANT LES
//*       VARIANTES D'INSTALLATION (S'IL Y EN A), EN DEMANDANT  SI
//*       NECESSAIRE LA SELECTION DE PORTIONS DE JCL D'INSTALLATION
//*       (MODULES DE JCL), EN DONNANT DES VALEURS APPROPRIEES AUX
//*       PARAMETRES D'INSTALLATION, ET EN PRECISANT EVENTUELLEMENT
//*       DES LIGNES A AJOUTER EN TETE OU EN FIN DE CHAQUE MODULE
//*       DE JCL.
//*********************************************************************
//*
//*  --- CREATION OF INSTALLATION JCL THROUGH 'MM1JCL'
//*      ****************************************
//*
//*      .MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ENTERING THE
//*       INSTALLATION VARIANTS (IF ANY), 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 ---------   MEMBRE DE PDS OU FICHIER SEQUENTIEL
//*                       RECEVANT LE JCL (LRECL=80)
//*                       PDS MEMBER OR SEQUENTIAL FILE RECEIVING
//*                       THE INSTALLATION JCL (LRECL=80)
//SYSIN DD *
===SELL <>               .LANGUAGE SELECTION (ENG OR FRA)
===SELV SQL              .PACBASE DB2 SQL
                         .SQL DB2 PACBASE
===SELV PEI              .PACBASE PRODUCTION ENVIRONMENT INTERFACE
                         .PACBASE ENVIRONNEMENT PRODUCTION
===SELV TAB              .PACTABLES
===SELV DSM              .DSMS
===SELV SEC              .PACBASE SECURITY INTERFACE OPTION
                         .PACBASE INTERFACE SYSTEME DE SECURITE
===SELV ROLD             .RETRIEVAL OF OLD PACBASE
                         .REPRISE ANCIENS PACBASE
===SELV PUF              .MODULE PUF-TP
                         .PUF-TP FUNCTION
===SELV S2K              .MODULE Pac/Impact
                         .Pac/Impact
===SELV PACTR            .MODULE PAC/TRANSFER
                         .PAC/TRANSFER FACILITY
===SELV VISP             .VISUALAGE SMALLTALK - PACBASE BRIDGE
                         .PONT VISUALAGE SMALLTALK - PACBASE
===SELV TEAM             .VA PAC - TEAMCONNECTION BRIDGE
                         .PONT VA PAC - TEAMCONNECTION
===PRM PRFJ=PACIN        .JOB NAMES PREFIXES (MAX OF 5 CHARACTERS)
                         .PREFIXE NOMS DE JOBS (5 CARACTERES MAXI)
===PRM CCPT=<>           .JOB ACCOUNTING CODES (JOB CARDS)
                         .CODE COMPTABLE JOBS (CARTES JOB)
===PRM CLASSJ=1          .JOB EXECUTION CLASS (JOB CARDS)
                         .CLASSE D'EXECUTION DES JOBS (CARTES JOB)
===PRM MSGCL=A           .JCL OUTPUT CLASS (MSGCLASS)
                         .CLASSE DE SORTIE DU JCL (MSGCLASS)
===PRM INDSV='EXP.PAC25' .PACBASE SYSTEM VSAM FILES INDEXES
                         .INDEX FICHIERS SYSTEME VSAM
===PRM INDSN='EXP.PAC25' .PACBASE SYSTEM NON VSAM FILES INDEXES
                         .INDEX FICHIERS SYSTEME NON VSAM
===PRM INDUV='CICS.PAC25' .PACBASE USER VSAM FILES INDEXES
                         .INDEX FICHIERS UTILISATEUR VSAM
===PRM INDUN='PAC.R25'   .PACBASE USER NON VSAM FILES INDEXES
                         .INDEX FICHIERS UTILISATEUR NON VSAM
===PRM ROOT='PB'         .ROOT OF SYSTEM (2 CHARACTERS)
                         .RADICAL DU SYSTEME (2 CARACTERES)
===PRM FILE='25'         .NUMBER OF USER'S PACBASE DATA BASE (2 CHAR.)
                         .NUMERO BASE UTILISATEUR (2 CARACTERES)
===PRM OUT=A             .JOB SYSOUT CLASS
                         .CLASSE DE SYSOUT DANS LES JOBS
===PRM U3480=3480        .CARTRIDGE UNIT
                         .UNIT CARTOUCHE
===PRM ICART=<>          .INSTALLATION CARTRIDGE VOL=SER=
                         .VOL=SER= CARTOUCHE INSTALLATION
===PRM UWK=SYSDA         .WORK FILES UNIT
                         .UNITE FICHIERS DE TRAVAIL
===PRM UNITSN=SYSDA      .NON VSAM PACBASE SYSTEM FILES UNIT
```

```
                          .UNITE FICHIERS PACBASE NON VSAM
    ===PRM UNITUN=SYSDA   .NON VSAM USER FILES UNIT
                          .UNITE FICHIERS UTILISATEUR NON VSAM
    ===PRM VOLSN=         .PACBASE SYSTEM NON VSAM FILES VOL=SER=
                          .VOL=SER= FICHIERS SYSTEME NON VSAM
    ===PRM VOLSV=         .PACBASE SYSTEM VSAM FILES VOLUME
                          .VOLUME FICHIERS SYSTEME VSAM
    ===PRM VOLUN=         .PACBASE USER NON VSAM FILES VOL=SER
                          .VOL=SER= FICHIERS UTILISATEUR NON VSAM
    ===PRM VOLUV=         .PACBASE USER VSAM FILES VOLUME
                          .VOLUME FICHIERS UTILISATEUR VSAM
    ===PRM VCAT=          .PACBASE USER FILES VSAM CATALOG
                          .CATALOGUE VSAM FICHIERS UTILISATEUR
    ===PRM SCAT=          .PACBASE SYSTEM FILES VSAM CATALOG
                          .CATALOGUE VSAM FICHIERS SYSTEME
    ===PRM DSCB='PAC.DSCB'    .DSCB MODEL FILE DSNAME
                              .DSNAME FICHIER DSCB MODELE
    ===PRM RADP='PB25'        .PREFIX OF PROCEDURE NAMES (MAXI 4 CHAR.)
                              .PREFIXE NOMS DES PROCEDURES (4 CAR. MAXI)
    ===PRM BIBP='SYS1.PROCLIB'     .PROCEDURE LIBRARY
                                   .BIBLIOTHEQUE DES PROCEDURES
    ===PRM MODB='PAC.R25.MBR8'     .BATCH LOAD MODULES LIBRARY
                                   .BIBLI LOAD MODULES BATCH
    ===PRM MODT='PAC.R25.MTR8'     .ON-LINE LOAD MODULES
                                   .BIBLI LOAD MODULES TP
    ===PRM MCVS='PAC.R25.COBVS'    .COBOL/VS LOAD MODULES
                                   .PROGRAMMES COBOL/VS
    ===PRM BIBT='SYS1.SORTLIB'     .SORT LIBRARY
                                   .BIBLIOTHEQUE DE  TRI
    ===PRM BCOB='SYS1.COB2LIB'     .COBOL ROUTINE LIBRARY
                                   .BIBLIOTHEQUE COBOL
    ===PRM CSDL='CICS330.SDFHLOAD' .DFHCSDUP STEPLIB DSN
                                   .DSN STEPLIB DFHCSDUP
    ===PRM DFHCSD='PAC.DFHCSD'     .PACBASE CSD DSN
                                   .DSN CSD PACBASE
    ===PRM GROUF='PACBASE'         .'GROUP' NAME FOR FILE
                                   .NOM DE "GROUP" POUR FICHIER
    ===PRM GROUP='PACBASE'         .'GROUP' NAME FOR PROGRAM
                                   .NOM DE "GROUP" POUR PROGRAMME
    ===PRM GROUT='PACBASE'         .'GROUP' NAME FOR TRANSACTION
                                   .NOM DE "GROUP" POUR TRANSACTION
    ===PRM LIST=<>                 .'LIST' NAME
                                   .NOM DE "LIST"
    ===PRM TABTDF=<>     .IF ===SELV TAB : DSN OF TABLES DESCRIPTION FILE
                         .SI ===SELV TAB : DSN DESCRIPTIFS TABLES
    ===PRM DSMS=<>       .IF ===SELV DSM : DSN OF PACBASE ELEMENTS (DC)
                         .SI ===SELV DSM : DSN ELEMENTS PACBASE (DC)
    ===BEGMOD
    ./ ADD NAME=$MODULE
    /*
    //
```

4.3.2. INSTALLATION OF THE COMPLETE JCL

INSTALLATION PREPARATION

INSTALLATION OF THE COMPLETE JCL

The MM1JCL module reads the JCL skeleton file (label 03) and produces a complete JCL. It allows you to:

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

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

. Select the installation variants to generate the JCL needed for specific processing, depending on site and installation conditions,

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

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

USER INPUT

Refer to the following paragraphs:

```
.Coding of MM1JCL commands
.Installation variants
.JCL modules
.JCL parameters
.JCL before/after lines
```

OUTPUT RESULT: COMPLETE JCL

The resulting SYSUT2 file contains all the installation and operation JCLs. This file may be modified (if necessary) via a text editor before beginning the installation.

Two operations must be performed on the complete JCL:

1. Global modifications (if necessary):

Adaptations can be performed on all the JCLs.

VSAM catalogues are entered as comments in the installation JCL:

```
- in the DELETE/DEFINE, as:     /*: CATALOG ($VCAT) */
                        or:     /*: CATALOG ($SCAT) */

- in the JCL STEPCAT's  as:     //*:STEPCAT  DD
                        and/or: //*:         DD

- in the procedure parameters as: //*:      VSAMCAT='$VCAT'
                              or: //*:      SYSTCAT='$SCAT'
```

When these parameters are not required, the resulting JCL is OK as it is.

When these parameters are required, affected lines should be changed into command lines. This is accomplished by:

- Transforming all '//*:' into '//',
- Substituting blanks for '/*:' and '*/'.

Blocking factors for large files can also be changed. Refer to paragraph 'Note on the files' Cisize/Blksize' (VA Pac only).

CAUTION:

. If the SMS product is installed, you should delete IDCAMS definition DD //GDGMOD lines in the installation JCLs with GenerationDataGroup allocation.

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

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

2. JCL splitting

In front of 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).

This allows for the complete JCL to be split in as many members as there are JCL modules in a PDS. The completed JCL file is to be used as SYSIN for the PDS update utility: IEUBUPDTE.

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

MM1JCL produces a list for each JCL module created, including parameters taken into account and according to required variants.

Since the JCL skeleton parameters are in the $xxxx format, during execution, if MM1JCL encounters a $ character that 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 the '$' in the flow processed by MM1JCL which are NOT parameters (in particular, PACDESIGN transactions).

CODING OF MM1JCL COMMANDS

```
===SELL lll               .Selection of installation
                           language:
                           lll = ENG (English)
                                 FRA (French)

===SELV vvvv              .Selection of variant
                           vvvv = variant code

===SELM mmmm1 mmmm2 ...   .Selection of JCL modules
                           mmmm1 = name of JCL module
                           mmmm2 = name of JCL module
                           etc.
                           The absence of a ===SELM line
                           involves the selection of all
                           JCL modules.

===PRM  PPPP=pppp         .Parameter
                           PPPP = name of parameter
                           pppp = value of parameter

NOTE: 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         )
```

4.3.3. DEFAULT INSTALLATION OPTIONS

DEFAULT INSTALLATION OPTIONS

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

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

.PARAMETERS (===PRM):

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

.MODULES (===SELM):

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

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

A line: ./ ADD NAME=$MODULE

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

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

NOTE ON THE FILES' "CISIZE/BLKSIZE"

For most files, the user does not need to modify the blocks, whatever the disk units (3380 or 3390). This applies to the CISIZE's of the VSAM files and BLKSIZE's of small files.

For important files, and since blocks are calculated for the optimization of disk space on 3380 units, if the site's disk units are of the 3390 type, the following can be changed:

```
LRECL=155,BLKSIZE=27280  to  LERCL=155,BLKSIZE=23405

LRECL=80,BLKSIZE=12560   to  LERCL=80,BLKSIZE=11440

LRECL=55,BLKSIZE=12595   to  LERCL=55,BLKSIZE=11440

LRECL=308,BLKSIZE=27104  to  LERCL=308,BLKSIZE=223408

LRECL=149,BLKSIZE=27267  to  LERCL=149,BLKSIZE=23393

LRECL=167,BLKSIZE=27555  to  LERCL=167,BLKSIZE=23714

LRECL=55,BLKSIZE=27280   to  LERCL=55,BLKSIZE=23430

LRECL=173,BLKSIZE=27161  to  LERCL=173,BLKSIZE=23355

LRECL=54,BLKSIZE=27270   to  LERCL=54,BLKSIZE=23436

In the GPRT, GPRP and GPRU procedures, change
BLKSIZE=27180   to  BLKSIZE=23400
```

The modifications should be done on the whole of the resulting JCL file, before splitting it in several members, so as to avoid doing them several times.

4.3.4. INSTALLATION VARIANTS

INSTALLATION VARIANTS

===SELV VVV .Comments

VVV	MEANING	COMMENTS
SQL	SQL Relational DBD Installation of the SQL catalog update COBOL source	VA Pac function
PEI	Production Environment Interface PEI file installation and recognition in VA Pac	VA Pac function
TAB	Pactables Installation of the VA Pac - Pactables interface components	VA Pac function
DSM	DSMS Installation of the VA Pac - DSMS interface components	VA Pac function
PACTR	Pac/Transfer	VA Pac function
PUF	PUFTP	VA Pac function
S2K	Pac/Impact	VA Pac function
VISP	VisualAge Smalltalk-VA Pac	VA Pac function
TEAM	Teamconnection-VA Pac	VA Pac function
SEC	Security Systems Interface Installation of the security system access sub-programs (RACF or TOPSECRET)	VA Pac function
ROLD	Retrieval of 802v02, 1.2, 1.5, 1.6, or 2.0 databases	Users of release 802v02 to 2.0

4.3.5. JCL MODULES

JCL MODULES

```
===SELM mmmm1 mmmm2 ... mmmmn
-------------------------------------------------------------
!  mmmm    ! Contents                                ! Nature!
!----------!-----------------------------------------!-------!
!          !                                         !       !
!          ! CICS UPDATE: CSDnxxxx                   !       !
!          !                                         !       !
! CSD1SYST ! CSD update -  VA Pac System components!  JCL  !
! CSD2BASE ! CSD update -  VA Pac Database          !   -   !
!          !               components                !       !
!          !                                         !       !
!          ! INSTALLATION                            !       !
!          !  VA Pac SYSTEM: Dnnxxxx                 !       !
!          !                                         !       !
! D01ALLOC ! Allocation of parameter PDSs            !  JCL  !
!          !                                         !       !
! D02CPAR  ! Loading of common parameters            !   -   !
! D02DIC   ! Loading of Dictionary complements       !   -   !
! D02DICVG ! Allocation and loading of the VINS      !   -   !
!          ! procedure input                         !       !
! D02METH  ! Loading of Entities and parameters of   !   -   !
!          ! the PACDESIGN methodologies             !       !
!          !                                         !       !
! D03DEFIN ! VSAM common file definition             !   -   !
! D03GDG   ! Definition of GDG for parameter backup  !   -   !
! D03INI   ! System file definitions                 !   -   !
!          !                                         !       !
! D04MBR   ! Batch program library allocation and    !   -   !
!          ! loading (MBR8)                          !       !
! D04MTR   ! On-line program library allocation      !   -   !
!          ! and loading (MTR8)                      !       !
! D04VS    ! COBOL/VS PAF sub-program library        !   -   !
!          ! allocation and loading                  !       !
!          !                                         !       !
! D05IPROC ! PROCLIB allocation (optional)           !   -   !
! D05PROC  ! Loading of batch procedures in the      !   -   !
!          ! PROCLIB                                 !       !
!          !                                         !       !
! D06SKEL  ! Loading of skeleton files               !   -   !
!          !                                         !       !
! D07AE0   ! Loading of error messages and           !   -   !
!          ! documentation                           !   -   !
! D07PARM  ! Installation of User Parameters         !       !
!          ! (AE and AP files, PARM procedure)       !       !
!          !                                         !       !
! D99INSL  ! List of components' dates               !       !
-------------------------------------------------------------
```

mmmm	Contents	Nature
	VA PAC SYSTEM OPTIONAL-PDS	
	INSTALLATION: Ennxxxx	
E01MSO	Allocation and loading of the multi-screen OLSD variant sources ("ZAR980")	JCL
E01SQL	Allocation and loading of SQL sources of DB2 catalog access programs (SQL or DB2 variants)	-
E02SECR	Loading of the Security System Interface sub-programs (SEC variant) in an Authorized Library (RACF only)	-
E02SECT	Loading of the Security System Interface sub-programs (SEC variant): sources of the sub-programs for access to the TOPSECRET tables to be compiled on the site (TOPSECRET only)	-
	VA PAC DATABASE INSTALLATION: Innxxxx	
I01SY	Database parameter PDS allocation	JCL
I02SY	Database parameter loading	-
I03DEF	Database VSAM file definitions	-
I03INI	GDG definitions and initializations PACBASE test database loading	-
I04REST	Test database restoration (REST procedure)	-
I05REAG	Initialization of database Generation-Print Request file (REAG procedure)	- -
I06PEI	Initialization of the PEI function files (PEI variant, SVPE and RSPE procedures)	-
I11VGE	Installation of the VISUAL Dictionary (VINS procedure)	-

mmmm	Contents		Nature
	INSTALLATION COMPLEMENTS		
	Pac/Impact		
	(S2K variant)		
I20GDGI	GDG Initialization		JCL
	Pac/Transfer facility		
	(PACTR variant)		
I30INIUV	Parameter-file initialization		JCL
I31INIJT	Compressed journal initialization		-
	VA Pac - TeamConnection		
	(TEAM variant)		
I40INITS	Parameter-file initialization		JCL
I41TEAM	Installation of the TeamConnection		-
	dictionary (VINS procedure)		
	PROCEDURE TEST JCL MODULES		
JCLUPDT	Batch update	UPDT	JCL
JCLGPRT	Generation and printing	GPRT	JCL
JCLARCH	Archival of journal	ARCH	JCL
JCLSAVE	Backup of the database	SAVE	JCL
JCLSVAG	Backup of AG file	SVAG	JCL
JCLMLIB	Library management	MLIB	JCL
JCLREOR	Reorganization of database	REOR	JCL
JCLREAG	Reorganization of AG file	REAG	JCL
JCLREST	Database restoration	REST	JCL
JCLEXLI	Library extraction	EXLI	JCL
JCLEXTR	Extraction of an entity	EXTR	JCL
JCLEXPJ	Transaction extraction	EXPJ	JCL
	using the journal file		
JCLUXSR	Library extraction	UXSR	-

mmmm	Contents	Procedure	Nature
	Pac/Impact (S2K)		
JCLIANA	Impact analysis	IANA	JCL
JCLINFP	FP initialization	INFP	-
JCLINFQ	FQ initialization	INFQ	-
JCLIPEP	Search-criteria printout	IPEP	-
JCLIPFQ	FQ file printout	IPFQ	-
JCLIPIA	Results printout	IPIA	-
JCLISEP	Entry point selection	ISEP	-
JCLISOS	String, operator selection	ISOS	-
	Pac/Transfer facility (PACTR)		
JCLTRDU	Production of DSMS transactions	TRDU	JCL
JCLTRJC	Compression of Journal file	TRJC	-
JCLTRPF	Generation of transfer transact.	TRPF	-
JCLTRRP	Extraction from Journal	TRRP	-
JCLTRRT	UV file retrieval	TRRT	-
JCLTRUP	Parameter update	TRUP	-
	VA Smalltalk / VA Pac Bridge (VISP)		
JCLVDWN	Download VA Pac/VA Smalltalk	VDWN	JCL
JCLVPUR	VA Smalltalk/VA Pac entity purge	VPUR	-
JCLVUP1	VA Smalltalk/VA Pac code upload	VUP1	-
JCLVUP2	VA Smalltalk/VA Pac transaction upload	VUP2	-
	FORMER RELEASE RETRIEVAL (ROLD)		JCL to be adapted
RPJ160	Journal retrieval	PJ16	
RPG200	PG file retrieval	RPPG	

4.3.6. JCL PARAMETERS

JCL PARAMETERS

```
===PRM PPPP=pppp                .Comments
```

- Parameter values which contain special characters must be entered between simple quotes.

- Comments on ===PRM lines must not exceed column 72, and must be preceded by a period (.).

NOTE: those parameters which have either the default value or the '< >' value MUST be set.

CODE PPPP	MEANING	DEFAULT pppp
	JOB CARDS	

PRFJ	Jobname prefix (Max.: 5 characters)	PAC
CCPT	Job accounting code	<>
CLASSJ	Job execution class	1
MSGCL	JCL output class	A
	CODING OF FILES' DSNs	

	VA Pac system files:	
	&INDSV.rrrrss: VSAM files	
	&INDSN.rrrrss: VSAM files	
	VA Pac user files:	
	&INDUV.rrffss: VSAM files	
	&INDUN.rrffss: Non-VSAM files	
	IND-- File name indices:	
	(24 characters max.)	
INDSV	VSAM, system	EXP.PAC12
INDSN	NON-VSAM, system (SAM, PDS)	EXP.PAC12
INDUV	VSAM, user	CICS.PAC12
INDUN	NON-VSAM, user (SAM)	PAC.R12

JCL PARAMETERS (Continued)

CODE PPPP	MEANING	DEFAULT pppp
	rr=ROOT, ff=FILE, ss=file code suffix	
ROOT	Root of the VA Pac system (2 characters other than 'PW')	PB
FILE	VA Pac Database number (2 characters other than 99, EE, EF)	80
OUT	SYSOUT print class	A
U3480	UNIT of installation tape/cartridge copy of the product-supplied media.	TAPE
ICART	VOL=SER installation media, copy of the product-supplied media.	<>
UWK	UNIT of work files used.	SYSDA
UNITSN	UNIT of non-VSAM system files.	SYSDA
UNITUN	UNIT of non-VSAM user files.	SYSDA
VOLSN	VOL=SER of non-VSAM system files.	
VOLSV	VOL=SER of VSAM system files.	
VOLUN	VOL=SER of non-VSAM user files.	
VOLUV	VOL=SER of VSAM user files.	
VCAT	VSAM catalog of the VA Pac database (User files)	
SCAT	VSAM catalog of the VA Pac system (System files)	
DSCB	DSNAME of DSCB model file used for files at generation.	'PAC.DSCB'
RADP	Prefix of names of procedures to install (maximum: 4 characters)	PB12
BIBP	DSNAME of the procedure library (PROCLIB)	'SYS1.PROCLIB'

JCL PARAMETERS (Continued)

CODE PPPP	MEANING	DEFAULT pppp
MODB	DSNAME of the batch load-module library	'PAC.R12.MBR8'
MODT	DSNAME of the on-line load-module library	'PAC.R12.MTR8'
MCVS	DSNAME of the COBOL/VS sub-program library	'PAC.R12.COBVS'
BIBT	DSNAME of the sort library (SORTLIB)	'SYS1.SORTLIB'
BCOB	DSNAME of COBOL II routine library	'SYS1.COB2LIB'
	UPDATE OF THE CICS CDS:	
CSDL	DFHCSDUP STEPLIB DSN	'CICS311.LOADLIB'
DFHCSD	VA Pac CSD DSN for update	'PAC.DFHCSD'
GROUP	VA Pac CSD Input Group programs	PACBASE
GROUF	files	PACBASE
GROUT	transactions	PACBASE
LIST	List in which the VA Pac group is added.	<>
	Pactables (if ===SELV TAB)	
TABTDF	Table description file DSN	<>
	DSMS (if ===SELV DSM)	
DSMS	VA Pac-element file DSN	<>

4.3.7. JCL BEFORE/AFTER LINES

JCL BEFORE/AFTER LINES

```
===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 in the MM1JCL 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 MM1JCL 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.

4.4. INSTALLATION PROCESS

INSTALLATION PROCESS

Once all the JCL modules are created, you may proceed with the installation of the VA Pac system. This installation is broken down into the following stages:

0. CICS update

0.1. VA Pac system components

0.2. VA Pac database components

1. VA Pac system installation

1.1. Initial allocation of parameter and Dictionary complement PDSs

1.2. Loading of parameter and Dictionary complement PDSs

1.3. VSAM and GDG file definitions

1.4. Allocation and loading of load-module libraries

1.5. Allocation and loading of batch procedures

1.6. Loading of generation skeleton files

1.7. Loading of error messages, documentation and user parameters

End. List of programs and installation files

2. VA Pac system installation optional complements

2.1. Allocation and loading of the OLSD multi-screen variant sources

2.2. Allocation and loading of source programs for access to the DB2 catalog of the SQL option (SQL variant)

2.3. Allocation and loading of the Security System Interface sub-programs (SEC variant)

3. Installation of the VA Pac database

3.1. Initial allocation of the parameter PDS

3.2. Loading of the parameter PDS

3.3. Initialization of the database files

3.4. Test database restoration

3.5. Initialization of the print-generation request file

3.6. Initialization of the Production Environment Interface files (PEI variant)

4. VA Pac Database optional installation components

4.1. Installation of the VA Smalltalk dictionary

4.2. Installation of the PAF function dictionary

4.3. Installation of the PAF+ Extension

4.4. Installation of the WorkStation mainframe components

4.5. Installation of the PACBENCH QUALITY CONTROL function (PQC) User Entity

5. CICS modifications

5.1. Specification of load-module libraries in the CICS submission JCL

5.2. Modification of the VA Pac transaction output (optional)

6. Other installation complements

6.1. Installation of Pac/Impact

6.2. Installation of the Pac/Transfer Facility

6.3. Installation of the VA Pac-TeamConnection Bridge

4.4.1. CICS UPDATE

0. CICS UPDATE

0.1. VA PAC SYSTEM COMPONENTS

===MOD CSD1SYST

rr=ROOT, VA Pac system root, transaction code prefixes, programs and files. User-defined (default: ROOT=PB).

```
Files:

rrrrAE: Error messages, on-line documentation and user
        parameters accessed by on-line VA Pac
rrrrAP: User parameters accessed by batch VA Pac
rrrrPA: User PAF requests work file

Programs:

rrQnnn, rrPnnn or rrRnnn - for programs
rrNnnn, rrMnnn or rrOnnn - for maps
```

plus a few specific programs, always prefixed with rr (see Sub-Chapter 'On-line program library', in Chapter 'VISUALAGE PACBASE COMPONENTS').

Transaction: rrEF

This performs common operations on user parameters handled in the AE and AP files, as well as operations on the AB and AC files of the Environment Production Interface.

0.2. VISUALAGE PACBASE DATABASE COMPONENTS

===MOD CSD2BASE

ff=FILE, database number, complement of the transaction code and of the prefixes of file codes in the VA Pac database. User-defined.

```
Files:

rrffAN:  VA Pac  database, Index file,
rrffAR:  VA Pac  database, Data file,
rrffAJ:  Journal file of transactions,
rrffAG:  Generation-Printing request file.

Files related to the PEI function:

rrffAB:  Production environment accessed on-line,
rrffAC:  Production environment updated on-line.
```

Transaction: rrff

This performs common database operations such as reads and updates.

NOTES

The "Dynamic Backout" option is required for the AN, AR, AJ, AG and AP files (and for the AC file of the PEI function).

The estimated number of STRINGS, INDEX BUFFERS and DATA BUFFERS indicated are the minimum numbers necessary for the system. Refer to Subchapter 'EFFICIENCY ENHANCEMENT' in Chapter ENVIRONMENT.

```
//$PRFJ.C1 JOB ($CCPT),'PACBASE DFHCSDUP',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - CICSCSD                              *
//*    CICS: CSD BATCH UPDATE - VA Pac SYSTEM                         *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
*********************************************************************
*      VA Pac       - $ROOT. COMMON FILES                           *
*********************************************************************
DEFINE FILE($ROOT.$ROOT.AE) GROUP($GROUF)
DESCRIPTION(PACBASE HELP)
       DSNAME($INDSV..$ROOT.$ROOT.AE)
       STRINGS(2)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOT.$ROOT.AP) GROUP($GROUF)
DESCRIPTION(PACBASE BATCH PARAM)
       DSNAME($INDSV..$ROOT.$ROOT.AP)
       STRINGS(2)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOT.$ROOT.PA) GROUP($GROUF)
DESCRIPTION(PACBASE PAF)
       DSNAME($INDSV..$ROOT.$ROOT.PA)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(2) INDEXBUFFERS(1)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
*********************************************************************
*      VA Pac        - $ROOT. PROGRAMS                              *
*********************************************************************
DEFINE PROGRAM($ROOT.PA00) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA01) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA10) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA11) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA12) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA13) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA14) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA15) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA16) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA17) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA18) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA19) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA20) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA21) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA22) GROUP($GROUP)
===SEQ FOR PEI
DEFINE PROGRAM($ROOT.PA30) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA31) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA32) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA33) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA34) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PA35) GROUP($GROUP)
===SEQ
DEFINE PROGRAM($ROOT.PBND) GROUP($GROUP)
DEFINE PROGRAM($ROOT.PPHP) GROUP($GROUP)
DEFINE PROGRAM($ROOT.QA00) GROUP($GROUP)
DEFINE PROGRAM($ROOT.QB00) GROUP($GROUP)
DEFINE PROGRAM($ROOT.QC00) GROUP($GROUP)
DEFINE PROGRAM($ROOT.QC01) GROUP($GROUP)
DEFINE PROGRAM($ROOT.QC50) GROUP($GROUP)
DEFINE PROGRAM($ROOT.QD00) GROUP($GROUP)
```

```
    DEFINE PROGRAM($ROOT.QE00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QF00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QF10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QG00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QH00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QH01) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QH20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QH30) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI01) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI02) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI03) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI04) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI05) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI21) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QI50) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QK10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QK20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QK30) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL21) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL30) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL40) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL41) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL45) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QL46) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QM00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP01) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP02) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP03) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP04) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP05) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP06) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP07) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP08) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QP50) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QR00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS02) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS03) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS04) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS05) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS06) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS08) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS09) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QS10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QT00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QT10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QT20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QT50) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QUPA) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QUPF) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QU00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QU01) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QU10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QU20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QV10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QV20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QV30) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QX00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QX01) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY01) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY02) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY03) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY04) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY05) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY10) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY11) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY20) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QY30) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.QZ00) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q000) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q100) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q101) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q102) GROUP($GROUP)
```

```
    DEFINE PROGRAM($ROOT.Q103) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q104) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q200) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q210) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q300) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q400) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q500) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q600) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q700) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q800) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.Q900) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.RS12) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R000) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R005) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R100) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R200) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R400) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R500) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R600) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.R980) GROUP($GROUP)
                               RESIDENT(YES)
    DEFINE PROGRAM($ROOT.TPST) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.TPWS) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.UCTR) GROUP($GROUP)
                               RESIDENT(YES)
    ===SEQ FOR SEC
    DEFINE PROGRAM($ROOT.SECT) GROUP($GROUP)
    ===SEQ FOR PUF
    DEFINE PROGRAM($ROOT.FANM) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FBIB) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FCTL) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FCOM) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FDBD) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FECR) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FENU) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FFOG) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FMCL) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FPGM) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FRUB) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FSDO) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FTXT) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.FVER) GROUP($GROUP)
    DEFINE PROGRAM($ROOT.F000) GROUP($GROUP)
    ===SEQ
    ************************************************************************
    *     VA Pac         - $ROOT.EE PARM-PEI TRANSACTION                   *
    ************************************************************************
    DEFINE TRANSACTION($ROOT.EE) GROUP($GROUT)
    DESCRIPTION(PACBASE PARM-PEI TRANSACTION) PROGRAM($ROOT.QUPA)
    ADD GROUP($GROUF) LIST($LIST)
    ADD GROUP($GROUP) LIST($LIST)
    ADD GROUP($GROUT) LIST($LIST)
    //
```

```
//$PRFJ.C2 JOB ($CCPT),'PACBASE DFHCSDUP',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - CICSCSD                              *
//*    CICS: CSD BATCH UPDATE FOR ONE VA Pac DATABASE                 *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
*********************************************************************
*      VA Pac        - $ROOT.$FILE. DATABASE FILES                  *
*********************************************************************
DEFINE FILE($ROOT.$FILE.AN) GROUP($GROUF)
DESCRIPTION(PACBASE INDEX)
       DSNAME($INDUV..$ROOT.$FILE.AN)
       STRINGS(2)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOT.$FILE.AR) GROUP($GROUF)
DESCRIPTION(PACBASE DATA)
       DSNAME($INDUV..$ROOT.$FILE.AR)
       STRINGS(2)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOT.$FILE.AG) GROUP($GROUF)
DESCRIPTION(PACBASE COMMAND)
       DSNAME($INDUV..$ROOT.$FILE.AG)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(2) INDEXBUFFERS(1)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOT.$FILE.AJ) GROUP($GROUF)
DESCRIPTION(PACBASE JOURNAL)
       DSNAME($INDUV..$ROOT.$FILE.AJ)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(2)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
===SEQ FOR PEI
DEFINE FILE($ROOT.$FILE.AB) GROUP($GROUF)
DESCRIPTION(PACBASE PEI)
       DSNAME($INDUV..$ROOT.$FILE.AB)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(2) INDEXBUFFERS(1)
       RECORDFORMAT(F)
       ADD(NO) BROWSE(YES) DELETE(NO) READ(YES) UPDATE(NO)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOT.$FILE.AC) GROUP($GROUF)
DESCRIPTION(PACBASE PEI)
       DSNAME($INDUV..$ROOT.$FILE.AC)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(2) INDEXBUFFERS(1)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
===SEQ
*********************************************************************
*     VA Pac         - $ROOT.$FILE. DATABASE TRANSACTION             *
*********************************************************************
DEFINE TRANSACTION($ROOT.$FILE) GROUP($GROUT)
DESCRIPTION(PACBASE $ROOT.$FILE TRANSACTION) PROGRAM($ROOT.R000)
```

```
//
```

4.4.2. SYSTEM - INITIAL ALLOCATION OF PARAMETER PDS

VA PAC SYSTEM INSTALLATION

1.1. INITIAL ALLOCATION OF PARAMETER AND DICTIONARY COMPLEMENTS PDS

```
System parameters:                  &INDSN..&ROOT.&ROOT.SY

Dictionary extensions & complements:&INDSN..&ROOT.&ROOT.SYDI

Dictionary entities - VINS input:   &INDSN..&ROOT.&ROOT.SYEN

PACDESIGN methodology entities:     &INDSN..&ROOT.&ROOT.SYMT


===MOD D01ALLOC      Job '$prfj.D1'

STEP1: IDCAMS: DELETE of files

STEP2: IEFBR14: allocation of files
```

```
//$PRFJ.D1   JOB ($CCPT),'PAC D01ALLOC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D01ALLOC                             *
//*    INITIAL ALLOCATION OF THE PARAMETER PDS AND ADDITIONAL FILES   *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOT.$ROOT.SY)
 DELETE ($INDSN..$ROOT.$ROOT.SYDI)
 DELETE ($INDSN..$ROOT.$ROOT.SYEN)
 DELETE ($INDSN..$ROOT.$ROOT.SYMT)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDSN..$ROOT.$ROOT.SY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(25,,5))
//SYDI     DD DSN=$INDSN..$ROOT.$ROOT.SYDI,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(100,,5))
//SYEN     DD DSN=$INDSN..$ROOT.$ROOT.SYEN,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=117,BLKSIZE=7020),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(7020,(25,,5))
//SYMT     DD DSN=$INDSN..$ROOT.$ROOT.SYMT,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(2000,,5))
//
```

4.4.3. SYSTEM - LOADING OF PARAMETER PDS

1.2. LOADING OF PARAMETER AND DICTIONARY-COMPLEMENT PDS

SYSTEM PARAMETERS: &INDSN..&ROOT.&ROOT.SY
```
===MOD D02CPAR       Job '$prfj.D2C'
```

STEP1: IEBUPDTE: loading of PDS members

CAUTION: replace all :/ with ./ before submitting the JOB.

DICTIONARY EXTENSION COMPLEMENTS:
&INDSN..&ROOT.&ROOT.SYDI
```
===MOD D02DIC        Job '$prfj.D2D'
```

STEP1: IEBUPDTE: Loading of the PDS members

CAUTION: replace all :/ with ./ before submitting the JOB.

STEP2: IEBCOPY: continuation of the loading

C.G.I. DICTIONARY ENTITIES (VINS procedure input):
&INDSN..&ROOT.&ROOT.SYEN
```
===MOD D02DICVG      Job '$prfj.D2E'
```

STEP1: IEBCOPY: Loading of the PDS members

PACDESIGN METHODOLOGY ENTITIES:
&INDSN..&ROOT.&ROOT.SYMT
```
===MOD D02METH       Job '$prfj.D2M'
```

STEP1: IEBCOPY: loading of PDS members

CAUTION: replace all :/ with ./ before submitting the JOB.

```
===FRM TYPE=DATA
//$PRFJ.D2C  JOB ($CCPT),'PAC D02CPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D02CPAR                              *
//*    LOADS PDS OF COMMON PARAMETERS                                 *
//*     .STEP1 : LOADING COMMON PARAMETERS                            *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..$ROOT.$ROOT.SY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/     ADD NAME=SRTREO1
 SORT FIELDS=(1,21,A,40,4,A,28,4,A,32,1,D,46,4,A,22,1,D),FORMAT=BI
:/     ADD NAME=SRTREO2
 SORT FIELDS=(1,55,A),FORMAT=BI
:/     ADD NAME=SRTQREO
 SORT FIELDS=(50,8,A),FORMAT=BI
:/     ADD NAME=SRTCPSN
 SORT FIELDS=(3,3,CH,A,6,7,CH,A)
:/     ADD NAME=SRTRVKE
 SORT FIELDS=(1,18,CH,A)
:/     ADD NAME=DF$ROOT.$ROOT.AE
 DELETE ($INDSV..$ROOT.$ROOT.AE) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.AE)   -
                  SHR (2,3) RUS   KEYS (12,0)     -
                  VOL ($VOLSV) -
                  CYL (10)     -
                  RECSZ (80 80) )   -
        INDEX   ( NAME ($INDSV..$ROOT.$ROOT.AE.I) -
                  CISZ (1024) )                   -
        DATA    ( NAME ($INDSV..$ROOT.$ROOT.AE.D) -
                  FSPC (2,1)                      -
        CISZ (4096) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DF$ROOT.$ROOT.AP
 DELETE ($INDSV..$ROOT.$ROOT.AP) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.AP)   -
                  SHR (2 3)   RUS   KEYS (7,0)    -
                  VOL ($VOLSV) -
                  CYL (2,1)    -
                  RECSZ (80,80) )                 -
        INDEX   ( NAME ($INDSV..$ROOT.$ROOT.AP.I) -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDSV..$ROOT.$ROOT.AP.D) -
                  FSPC (50,5)                     -
        CISZ (4096) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DF$ROOT.$ROOT.SC
 DELETE ($INDSV..$ROOT.$ROOT.SC) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.SC) -
                  SHR (2,3)  RUS   KEY (4 0)      -
                  VOL ($VOLSV)                    -
                  RECSZ (3204,3204) )             -
        INDEX   ( NAME ($INDSV..$ROOT.$ROOT.SC.I) -
                  CISZ (256) )                    -
        DATA    ( NAME ($INDSV..$ROOT.$ROOT.SC.D) -
                  FSPC (10,5) SPEED  TRK (5)      -
        CISZ (3584) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DF$ROOT.$ROOT.SG
 DELETE ($INDSV..$ROOT.$ROOT.SG) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.SG) -
                  SHR (2,3)  RUS   KEY (5 0)      -
                  VOL ($VOLSV)                    -
                  RECSZ (4605,4605) )             -
        INDEX   ( NAME ($INDSV..$ROOT.$ROOT.SG.I) -
                  CISZ (256) )                    -
        DATA    ( NAME ($INDSV..$ROOT.$ROOT.SG.D) -
                  FSPC (10,5) SPEED  CYL (3 1)    -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DF$ROOT.$ROOT.SR
 DELETE ($INDSV..$ROOT.$ROOT.SR) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.SR) -
                  SHR (2,3)  RUS   KEY (5 0)      -
```

```
                    VOL ($VOLSV)                    -
                    RECSZ (4605,4605) )             -
          INDEX   ( NAME ($INDSV..$ROOT.$ROOT.SR.I) -
                    CISZ (256) )                    -
          DATA    ( NAME ($INDSV..$ROOT.$ROOT.SR.D) -
                    FSPC (10,5) SPEED  TRK (5)      -
          CISZ (5120) )   /*: CATALOG ($SCAT) */
    :/     ADD NAME=DF$ROOT.$ROOT.SP
     DELETE ($INDSV..$ROOT.$ROOT.SP) CLUSTER
       DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.SP) -
                    SHR (2,3)  RUS   KEY (5 0)      -
                    VOL ($VOLSV)                    -
                    RECSZ (4605,4605) )             -
          INDEX   ( NAME ($INDSV..$ROOT.$ROOT.SP.I) -
                    CISZ (256) )                    -
          DATA    ( NAME ($INDSV..$ROOT.$ROOT.SP.D) -
                    FSPC (10,5) SPEED  TRK (3 1)    -
          CISZ (5120) )   /*: CATALOG ($SCAT) */
    :/     ADD NAME=DF$ROOT.$ROOT.GS
     DELETE ($INDUV..$ROOT.$ROOT.GS) CLUSTER
     DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$ROOT.GS)   -
                    SHR (2)   RUS   KEYS (25,0)   -
                    VOL ($VOLUV) -
                    CYL (1 1)    -
                    RECSZ (203,203) )               -
          INDEX   ( NAME ($INDUV..$ROOT.$ROOT.GS.I) -
                    CISZ (1024) )                   -
          DATA    ( NAME ($INDUV..$ROOT.$ROOT.GS.D) -
                    FSPC (10,5)                     -
          CISZ (4096) )         /*: CATALOG ($VCAT) */
    :/     ADD NAME=DF$ROOT.$ROOT.SS
     DELETE ($INDSV..$ROOT.$ROOT.SS) CLUSTER
       DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.SS) -
                    SHR (2,3)  RUS   KEY (5 0)      -
                    VOL ($VOLSV)                    -
                    RECSZ (4605,4605) )             -
          INDEX   ( NAME ($INDSV..$ROOT.$ROOT.SS.I) -
                    CISZ (256) )                    -
          DATA    ( NAME ($INDSV..$ROOT.$ROOT.SS.D) -
                    FSPC (10,5) SPEED  CYL (3 1)    -
          CISZ (5120) )   /*: CATALOG ($SCAT) */
    ===SEQ FOR TAB
    :/     ADD NAME=DFTABTDF
     DELETE ($TABTDF) CLUSTER
     DEFINE CLUSTER ( NAME ($TABTDF)                 -
                    SHR (2 3)   RUS   KEYS (21,0)   -
                    VOL ($VOLUV) -
                    CYL (1 1)    -
                    RECSZ (240,240) )               -
          INDEX   ( NAME ($TABTDF.I)                -
                    CISZ (1024) )                   -
          DATA    ( NAME ($TABTDF.D)                -
                    FSPC (10,5)                     -
          CISZ (2048) )         /*: CATALOG ($VCAT) */
    ===SEQ FOR DSM
    :/     ADD NAME=DFDSMSDC
     LISTCAT ENTRIES ($DSMS) CLUSTER
     IF LASTCC NE 0 THEN DO
     DELETE ($DSMS) CLUSTER
     DEFINE CLUSTER ( NAME ($DSMS)           -
                    SHR (2,3) REUSE  KEYS (31,2)        -
                    VOL ($VOLUV) -
                    CYL (1 1)    -
                    RECSZ (50,168) )                    -
          INDEX   ( NAME ($DSMS.I)          -
                    CISZ (1024) )                   -
          DATA    ( NAME ($DSMS.D)          -
                    FSPC (10,5)                     -
          CISZ (4096) )         /*: CATALOG ($VCAT) */
     END
    ===SEQ FOR TEAM
    :/     ADD NAME=DF$ROOT.$ROOT.TS
     DELETE ($INDSV..$ROOT.$ROOT.TS) CLUSTER
     DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.TS)   -
                    SHR (2,3) RUS   KEYS (14,1)     -
                    VOL ($VOLSV) -
                    CYL (10)     -
```

```
                  RECSZ (80 80) )   -
         INDEX   ( NAME ($INDSV..$ROOT.$ROOT.TS.I) -
                   CISZ (1024) )                   -
         DATA    ( NAME ($INDSV..$ROOT.$ROOT.TS.D) -
                   FSPC (10,5)                      -
         CISZ (4096) )   /*: CATALOG ($SCAT) */
 END
===SEQ
:/     ADD NAME=DF$ROOT.$ROOT.PA
 DELETE ($INDSV..$ROOT.$ROOT.PA) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..$ROOT.$ROOT.PA)   -
                  SHR (2 3)  RUS  KEYS (37 2)     -
                  VOL ($VOLSV) -
                  CYL (5 1)    -
                  RECSZ (200 539) )              -
         INDEX   ( NAME ($INDSV..$ROOT.$ROOT.PA.I) -
                   CISZ (512) )                    -
         DATA    ( NAME ($INDSV..$ROOT.$ROOT.PA.D) -
                   FSPC (10 5) SPEED               -
         CISZ (4096) )         /*: CATALOG ($SCAT) */
:/     ADD NAME=DFSYSPAF
 DELETE ($INDUV..SYSPAF.&USER) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..SYSPAF.&USER)     -
                  SHR (2 3)  RUS  KEYS(12 0)      -
                  VOL ($VOLUV) -
                  CYL (5 1)    -
                  RECSZ (170 468) )               -
         INDEX   ( NAME ($INDUV..SYSPAF.&USER.I)   -
                   CISZ (512) )                    -
         DATA    ( NAME ($INDUV..SYSPAF.&USER.D)   -
                   FSPC (10 5) SPEED               -
         CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFSYSEXT
 DELETE ($INDUV..SYSEXT.&USER) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..SYSEXT.&USER)     -
                  SHR (2 3)  RUS  KEYS(12 0)      -
                  VOL ($VOLUV) -
                  CYL (5 1)    -
                  RECSZ (112 112) )               -
         INDEX   ( NAME ($INDUV..SYSEXT.&USER.I)   -
                   CISZ (512) )                    -
         DATA    ( NAME ($INDUV..SYSEXT.&USER.D)   -
                   FSPC (10 5) SPEED               -
         CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFSYIANA
 DELETE ($INDUV..SYIANA.&USER) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..SYIANA.&USER)     -
                  SHR (2 3)  RUS  KEYS(94 0)      -
                  VOL ($VOLUV) -
                  CYL (5 1)    -
                  RECSZ (100 100) )               -
         INDEX   ( NAME ($INDUV..SYIANA.&USER.I)   -
                   CISZ (512) )                    -
         DATA    ( NAME ($INDUV..SYIANA.&USER.D)   -
                   FSPC (10 5) SPEED               -
         CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DLSYSPAF
 DELETE ($INDUV..SYSPAF.&USER) CLUSTER
:/     ADD NAME=DLSYSEXT
 DELETE ($INDUV..SYSEXT.&USER) CLUSTER
:/     ADD NAME=DLSYIANA
 DELETE ($INDUV..SYIANA.&USER) CLUSTER
:/     ADD NAME=MAXKEY
99999999999999999999999999999999999999999999999999999999999999999999
:/     ADD NAME=REPRO999
 REPRO INFILE (MAXKEY)   OUTFILE (SYSPAF)
:/     ADD NAME=REPRO
 REPRO INFILE (IN1)    OUTFILE (OU1)
:/     ADD NAME=VERIFAN
 VERIFY FILE (PAC7AN)
:/     ADD NAME=VERIFAR
 VERIFY FILE (PAC7AR)
:/     ADD NAME=VERIFAJ
 VERIFY FILE (PAC7AJ)
:/     ADD NAME=VERIFAG
 VERIFY FILE (PAC7AG)
:/     ADD NAME=VERIFAE
```

```
 VERIFY FILE (PAC7AE)
:/     ADD NAME=VERIFAP
 VERIFY FILE (PAC7AP)
:/     ADD NAME=VERIFSG
 VERIFY FILE (PAC7SG)
:/     ADD NAME=VERIFSC
 VERIFY FILE (PAC7SC)
:/     ADD NAME=VERIFSR
 VERIFY FILE (PAC7SR)
:/     ADD NAME=VERIFSP
 VERIFY FILE (PAC7SP)
:/     ADD NAME=VERIFSS
 VERIFY FILE (PAC7SS)
===SEQ FOR PEI
:/     ADD NAME=VERIFAB
 VERIFY FILE (PAC7AB)
:/     ADD NAME=VERIFAC
 VERIFY FILE (PAC7AC)
===SEQ FOR TAB
:/     ADD NAME=VERIFTD
 VERIFY FILE (PAC7TD)
===SEQ
:/     ADD NAME=VERIFEM
 VERIFY FILE (PAC7EM)
:/     ADD NAME=LI$ROOT.$ROOT.AE
 LISTCAT ENTRIES ($INDSV..$ROOT.$ROOT.AE) CLUSTER
PP
//
```

```
===FRM TYPE=DATA
//$PRFJ.D2D  JOB ($CCPT),'PAC D02DIC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D02DIC                               *
//*    LOADS PDS OF ADDITIONAL ENTITIES AND ELEMENTS                  *
//*    FOR THE DICTIONARY AND THE EXTENSIONS                          *
//*    (PAF, XPAF, XPDM, PQC)                                         *
//*     .STEP1 : LOADING JCL EXAMPLES (XPAF XPDM)                     *
//*     .STEP2 : LOADING ENTITIES AND ELEMENTS                        *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..$ROOT.$ROOT.SYDI,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
PP
//STEP2  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTDI    DD DSN=$INDSN..$ROOT.$ROOT.SYDI,DISP=SHR
//INPDI    DD DSN=PAC.SRCDIC,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(7,SL)
//SYSIN    DD *
  C I=((INPDI,R)),O=OUTDI
  S M=((PAFDICA,PAFDIC))
  S M=((PGDPA,PAFPTEX))
  S M=((PQCRENG,PQCRULE))
  S M=((PQCENG,PQCUPDT))
  S M=((EMUPDTA,EMUPDT))
//
```

```
//$PRFJ.D2E  JOB ($CCPT),'PAC D02DICVG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D02DICVG                             *
//*    LOADING OF DICTIONARY ENTITIES FOR VINS PROCEDURE              *
//*     .STEP1 : LOADING                                              *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTDI    DD DSN=$INDSN..$ROOT.$ROOT.SYEN,DISP=SHR
//INPDI    DD DSN=PAC.INSDIC,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(15,SL)
//SYSIN    DD *
  C I=((INPDI,R)),O=OUTDI
  S M=VGEDIC
  S M=TEAMDIC
//
```

```
//$PRFJ.D2M  JOB ($CCPT),'PAC D02METH',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D02METH                              *
//*    LOADING OF PACDESIGN METHOD ENTITIES' PDS                      *
//*     .STEP1 : LOADING                                              *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTMT    DD DSN=$INDSN..$ROOT.$ROOT.SYMT,DISP=SHR
//INPMT    DD DSN=PAC.SRCMTH,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(17,SL)
//SYSIN    DD *
  C I=((INPMT,R)),O=OUTMT
  S M=DESADM
  S M=((DESIFWA,DESIFW))
  S M=DESIFWP
  S M=((DESMERA,DESMER))
  S M=((DESOMTA,DESOMT))
  S M=DESYSM
  S M=((PARMADMA,PARMADM))
  S M=((PARMIFWA,PARMIFW))
  S M=((PARMMERA,PARMMER))
  S M=((PARMOMTA,PARMOMT))
  S M=((PARMYSMA,PARMYSM))
//
```

4.4.4. SYSTEM - VSAM AND GDG FILE DEFINITIONS

1.3. VSAM AND GDG FILE DEFINITIONS

DEFINITION OF SYSTEM FILES

```
===MOD D03DEFIN      Job '$prfj.D3'
```

STEP1: IDCAMS: DELETE/DEFINE

ALLOCATION OF MODEL DSCB - DEFINITION OF GDG

```
===MOD D03GDG        Job '$prfj.D3G'

STEP1: IDCAMS:       DELETE of model DSCB file
STEP2: IEFBR14:      Allocation of model DSCB
STEP3: IDCAMS:       Definition of GDG for user
                     parameter backup (PE)
STEP4: IEBGENER:     Initialization of the user
                     parameter backup (PE)
```

INITIALIZATION OF FILES

```
===MOD D03INI        Job '$prfj.D3I'

STEP1: IDCAMS   : Initialization of AP file max. key
                  (User parameters accessed in batch mode)

STEP2: IDCAMS   : Initialization of PA work file max. key
                  (PAF user on-line requests)

STEP3: IDCAMS   : Initialization of GS extraction schema
                  file max. key (PAF Extension)
```

```
//$PRFJ.D3  JOB ($CCPT),'PAC D03DEFIN',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D03DEFIN                             *
//*    DEFINITION OF COMMON VSAM FILES                                *
//*     STEP1 : DELETE/DEFINE                                         *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SC),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SG),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SR),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SP),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SS),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.AP),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.PA),DISP=SHR
//      DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.GS),DISP=SHR
//*
//
```

```
//$PRFJ.D3G JOB ($CCPT),'PAC D03GDG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//* VA Pac        INSTALLATION - D03GDG                               *
//*   JOB TO RUN ONLY THE FIRST TIME VA PAC IS INSTALLED              *
//*    . BUILDING OF DSCB MODEL                                       *
//*    . DEFINE GDG                                                   *
//*    . FILE INITIALIZATION (USER PARAMETER BACKUP)                  *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS            *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($DSCB)
//*
//STEP2  EXEC PGM=IEFBR14
//DSCB     DD DISP=(,CATLG),SPACE=(TRK,0),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         DSN=$DSCB
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//GDGMOD   DD DSN=$INDSN..$ROOT.$ROOT.PE,
//         DISP=(,KEEP,DELETE),
//         SPACE=(TRK,0),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         DCB=($DSCB,RECFM=FB,LRECL=80,BLKSIZE=3440)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDSN..$ROOT.$ROOT.PE) LIMIT (3) SCR)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//SYSUT2   DD DSN=$INDSN..$ROOT.$ROOT.PE(+1),
//         DISP=(,CATLG,DELETE),
//         SPACE=(TRK,(1,1),RLSE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         DCB=($DSCB,RECFM=FB,LRECL=80,BLKSIZE=3440)
//
```

```
//$PRFJ.D3I JOB ($CCPT),'PAC D03INI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              INSTALLATION - D03INI                                *
//*   INITIALIZATION: JOB TO RUN ONLY                                 *
//*   THE FIRST TIME VA PAC IS INSTALLED                              *
//*    . INITIALIZATION OF "AP" USER PARAMETER FILE                   *
//*    . INITIALIZATION OF P.A.F. WORK FILES                          *
//*    . INITIALIZATION OF PDM EXTENSION-USER FILE "GS"               *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//APO      DD DSN=$INDSV..$ROOT.$ROOT.AP,DISP=SHR
//API      DD *
999999999999
//SYSIN    DD *
  REPRO INFILE (API)   OUTFILE (APO)
//*
//STEP2  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//PAO      DD DSN=$INDSV..$ROOT.$ROOT.PA,DISP=SHR
//PAI      DD *
99999999999
//SYSIN    DD *
  REPRO INFILE (PAI)   OUTFILE (PAO)
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//GSO      DD DSN=$INDUV..$ROOT.$ROOT.GS,DISP=SHR
//GSI      DD *
999999999999999999999999
//SYSIN    DD *
  REPRO INFILE (GSI)   OUTFILE (GSO)
//*
//
```

4.4.5. SYSTEM - ALLOC./LOADING OF LOAD-MODULE LIBRARY

1.4. ALLOCATION AND LOADING OF THE LOAD-MODULE LIBRARIES

NOTE:

If the library BLKSIZE is not suitable, modify its value in the allocations steps (STEP2), and replace the C(opy) with COPYMOD in the loading steps (STEP3).

BATCH LOAD-MODULES

```
===MOD D04MBR     Job '$prfj.D4B'

STEP1: IDCAMS:    DELETE of the batch load-module library
STEP2: IEFBR14:   allocation of the library
STEP3: IEBCOPY:   loading of programs
```

ON-LINE LOAD-MODULES

```
===MOD D04MTR     Job '$prfj.D4O'
```

CAUTION: BE SURE TO SELECT THE UPPERCASE-LOWERCASE TRANSFORMATION PROGRAM (rrUCTR)

This program's installation is stated as comments. Before running the JOB, activate the relevant line, at the end of the SYSIN section of STEP3:

CICS ESA 3.1 ==> S M=((PWUCTR,rrUCTR))
CICS ESA 3.2 ==> S M=((PWUCTX,rrUCTR))
CICS ESA 3.3 ==> S M=((PWUCTZ,rrUCTR))

```
STEP1: IDCAMS:    DELETE of on-line load-module library
STEP2: IEFBR14:   allocation of the library
STEP3: IEBCOPY:   loading of maps and programs
```

COBOL/VS LOAD-MODULES

```
===MOD D04VS      Job '$prfj.D4V'
```

This installation is necessary only for those users who wish to develop COBOL/VS programs with the PAF function or its extension.

```
STEP1: IDCAMS:    DELETE of the on-line load-module library
STEP2: IEFBR14:   allocation of the library
STEP3: IEBCOPY:   loading of maps and programs
```

```
//$PRFJ.D4B JOB ($CCPT),'PAC D04MBR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D04MBR                               *
//*    INITIAL ALLOCATING AND LOADING OF THE MBR8 PDS                 *
//*    OF BATCH PROGRAMS                                              *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*     .STEP3 : LOADING                                              *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($MODB)
//*
//STEP2  EXEC PGM=IEFBR14
//LNKB     DD DSN=$MODB,
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6144,(2000,200,30)),DCB=(RECFM=U,BLKSIZE=6144)
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTB     DD DSN=$MODB,DISP=OLD                BATCH
//INB      DD DSN=PAC.MBR8,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(4,SL)
//SYSIN    DD *
  C   I=((INB,R)),O=OUTB
          S M=MM1JCL
          S M=PACA05
          S M=PACA10
          S M=PACA15
          S M=PACA20
          S M=PACA90
          S M=PACB
          S M=PACB30
          S M=PACB31
          S M=PACB40
          S M=PACB80
          S M=PACC30
          S M=PACC40
          S M=PACC80
          S M=PACDTP
          S M=PACD30
          S M=PACD40
          S M=PACD80
          S M=PACD90
          S M=PACE30
          S M=PACE40
          S M=PACE80
          S M=PACFMB
          S M=PACFGY
          S M=PACFTD
          S M=PACCTL
          S M=PACHOI
          S M=PACG3C
          S M=PACG3S
          S M=PACG4S
          S M=PACG8C
          S M=PACG8S
          S M=PACINS
          S M=PACK30
          S M=PACK80
          S M=PACK90
          S M=PACL30
          S M=PACL80
          S M=PACL90
          S M=PACL92
          S M=PACL93
          S M=PACL95
          S M=PACM30
          S M=PACM80
```

```
            S M=PACN25
            S M=PACN30
            S M=PACN35
            S M=PACN40
            S M=PACN50
            S M=PACN80
            S M=PACP30
            S M=PACP40
            S M=PACP80
            S M=PACP82
            S M=PACQ
   ===SEQ FOR PEI
            S M=PACR01
            S M=PACR10
            S M=PACR20
            S M=PACR22
            S M=PACR30
            S M=PACR40
            S M=PACR60
            S M=PACR61
            S M=PACR90
   ===SEQ
            S M=PACSEP
            S M=PACS30
            S M=PACS40
            S M=PACS50
            S M=PACSJO
            S M=PACS60
            S M=PACSPU
            S M=PACS75
            S M=PACS80
            S M=PACSRM
            S M=PACSMD
   ===SEQ FOR TAB
            S M=PACTIN
            S M=PACTI1
            S M=PACT40
            S M=PACT41
            S M=PACT45
            S M=PACT50
            S M=PACT51
   ===SEQ
            S M=PACU15
            S M=PACU80
            S M=PACU99
            S M=PACX
   ===SEQ FOR S2K
            S M=PAN200
            S M=PAN205
            S M=PAN210
            S M=PAN212
            S M=PAN215
            S M=PAN220
            S M=PAN230
            S M=PAN240
            S M=PAN250
            S M=PAN255
            S M=PAN260
            S M=PAN270
            S M=PAN280
   ===SEQ FOR DSM
            S M=PDS600
            S M=PDS610
   ===SEQ
            S M=PBBTST
            S M=PBBTWS
            S M=PBBT98
            S M=PADM10
            S M=PAFP10
            S M=PAF900
            S M=PREI00
            S M=PREI40
            S M=PREI50
            S M=PRE986
            S M=PRMSYS
            S M=PRUTMB
            S M=PTAR20
```

```
            S M=PTARSD
            S M=PTASVD
   ===SEQ FOR TEAM
            S M=PTC010
            S M=PTC030
            S M=PTC100
            S M=PTC200
            S M=PTC220
            S M=PTC400
            S M=PTC440
   ===SEQ
            S M=PTED30
            S M=PTED60
            S M=PTEP90
            S M=PTEXD0
            S M=PTEX30
            S M=PTEX80
            S M=PTUBAS
            S M=PTUCSS
   ===SEQ FOR PACTR
            S M=PTUG05
            S M=PTUG06
            S M=PTUG07
            S M=PTUG10
            S M=PTUG11
            S M=PTUG12
            S M=PTUG42
            S M=PTUG44
            S M=PTUG46
            S M=PTUG50
            S M=PTUG60
            S M=PTUG61
            S M=PTUG90
   ===SEQ
            S M=PTUESS
            S M=PTULOI
            S M=PTULVB
            S M=PTUN00
            S M=PTUN10
            S M=PTUN40
            S M=PTUQ10
            S M=PTUQ15
            S M=PTUQ20
            S M=PTUQ24
            S M=PTUQ25
            S M=PTUQ30
            S M=PTUR00
            S M=PTU001
            S M=PTU004
            S M=PTU100
            S M=PTU120
            S M=PTU130
            S M=PTU140
            S M=PTU2CL
            S M=PTU200
            S M=PTU208
            S M=PTU210
            S M=PTU220
            S M=PTU240
            S M=PTU300
            S M=PTU320
            S M=PTU380
            S M=PTU400
            S M=PTU402
            S M=PTU420
            S M=PTU500
            S M=PTU502
            S M=PTU550
            S M=PTU560
            S M=PTU630
            S M=PTU640
            S M=PTU810
            S M=PTU815
            S M=PTU850
            S M=PTU855
   ===SEQ FOR VISP
            S M=PVA100
```

```
            S M=PVA110
            S M=PVA300
            S M=PVA310
            S M=PVA320
            S M=PVA400
   ===SEQ
            S M=PYSMCC
            S M=PYSMC2
            S M=PYSMC3
            S M=UTIXSR
   ===SEQ FOR ROLD
            S M=PTU908
            S M=REP2PJ
   ===SEQ FOR SEC
            S M=PACSECB
   ===SEQ
   /*
   //
```

```
//$PRFJ.D4O JOB ($CCPT),'PAC D04MTR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              INSTALLATION - D04MTR                                *
//*    INITIAL ALLOCATING AND LOADING OF THE MTR8 PDS                 *
//*    OF ON-LINE PROGRAMS                                            *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*     .STEP3 : LOADING                                              *
//*  ******************************************************           *
//*  WARNING : LOWER CASE UPPER CASE TRANSLATION INSTALLATION         *
//*  =======   ACTIVATE THE CORRESPONDING SELECT LINE:                *
//*  ******************************************************           *
//*   CICS ESA 3.1  ==>  S M=((PWUCTR,$ROOT.UCTR))                    *
//*   CICS ESA 3.2  ==>  S M=((PWUCTX,$ROOT.UCTR))                    *
//*   CICS ESA 3.3  ==>  S M=((PWUCTZ,$ROOT.UCTR))                    *
//*                                                                   *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($MODT)
//*
//STEP2  EXEC PGM=IEFBR14
//LNKT     DD DSN=$MODT,
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6144,(1500,150,40)),DCB=(RECFM=U,BLKSIZE=6144)
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTT     DD DSN=$MODT,DISP=OLD                ON-LINE
//INT      DD DSN=PAC.MTR8,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(5,SL)
//SYSIN    DD *
  C   I=((INT,R)),O=OUTT
          S M=CALL
          S M=((PWTPST,$ROOT.TPST))
          S M=((PWTPWS,$ROOT.TPWS))
          S M=PBLTDLI
          S M=((PWPA00,$ROOT.PA00))
          S M=((PWPA01,$ROOT.PA01))
          S M=((PWPA10,$ROOT.PA10))
          S M=((PWPA11,$ROOT.PA11))
          S M=((PWPA12,$ROOT.PA12))
          S M=((PWPA13,$ROOT.PA13))
          S M=((PWPA14,$ROOT.PA14))
          S M=((PWPA15,$ROOT.PA15))
          S M=((PWPA16,$ROOT.PA16))
          S M=((PWPA17,$ROOT.PA17))
          S M=((PWPA18,$ROOT.PA18))
          S M=((PWPA19,$ROOT.PA19))
          S M=((PWPA20,$ROOT.PA20))
          S M=((PWPA21,$ROOT.PA21))
          S M=((PWPA22,$ROOT.PA22))
===SEQ FOR PEI
          S M=((PWPA30,$ROOT.PA30))
          S M=((PWPA31,$ROOT.PA31))
          S M=((PWPA32,$ROOT.PA32))
          S M=((PWPA33,$ROOT.PA33))
          S M=((PWPA34,$ROOT.PA34))
          S M=((PWPA35,$ROOT.PA35))
===SEQ
          S M=((PWPBND,$ROOT.PBND))
          S M=((PWPPHP,$ROOT.PPHP))
          S M=((PWQA00,$ROOT.QA00))
          S M=((PWQB00,$ROOT.QB00))
          S M=((PWQC00,$ROOT.QC00))
          S M=((PWQC01,$ROOT.QC01))
          S M=((PWQC50,$ROOT.QC50))
          S M=((PWQD00,$ROOT.QD00))
          S M=((PWQE00,$ROOT.QE00))
```

```
S M=((PWQF00,$ROOT.QF00))
S M=((PWQF10,$ROOT.QF10))
S M=((PWQG00,$ROOT.QG00))
S M=((PWQH00,$ROOT.QH00))
S M=((PWQH01,$ROOT.QH01))
S M=((PWQH20,$ROOT.QH20))
S M=((PWQH30,$ROOT.QH30))
S M=((PWQI00,$ROOT.QI00))
S M=((PWQI01,$ROOT.QI01))
S M=((PWQI02,$ROOT.QI02))
S M=((PWQI03,$ROOT.QI03))
S M=((PWQI04,$ROOT.QI04))
S M=((PWQI05,$ROOT.QI05))
S M=((PWQI20,$ROOT.QI20))
S M=((PWQI21,$ROOT.QI21))
S M=((PWQI50,$ROOT.QI50))
S M=((PWQK10,$ROOT.QK10))
S M=((PWQK20,$ROOT.QK20))
S M=((PWQK30,$ROOT.QK30))
S M=((PWQL10,$ROOT.QL10))
S M=((PWQL20,$ROOT.QL20))
S M=((PWQL21,$ROOT.QL21))
S M=((PWQL30,$ROOT.QL30))
S M=((PWQL40,$ROOT.QL40))
S M=((PWQL41,$ROOT.QL41))
S M=((PWQL45,$ROOT.QL45))
S M=((PWQL46,$ROOT.QL46))
S M=((PWQM00,$ROOT.QM00))
S M=((PWQP00,$ROOT.QP00))
S M=((PWQP01,$ROOT.QP01))
S M=((PWQP02,$ROOT.QP02))
S M=((PWQP03,$ROOT.QP03))
S M=((PWQP04,$ROOT.QP04))
S M=((PWQP05,$ROOT.QP05))
S M=((PWQP06,$ROOT.QP06))
S M=((PWQP07,$ROOT.QP07))
S M=((PWQP08,$ROOT.QP08))
S M=((PWQP50,$ROOT.QP50))
S M=((PWQR00,$ROOT.QR00))
S M=((PWQS02,$ROOT.QS02))
S M=((PWQS03,$ROOT.QS03))
S M=((PWQS04,$ROOT.QS04))
S M=((PWQS05,$ROOT.QS05))
S M=((PWQS06,$ROOT.QS06))
S M=((PWQS08,$ROOT.QS08))
S M=((PWQS09,$ROOT.QS09))
S M=((PWQS10,$ROOT.QS10))
S M=((PWQT00,$ROOT.QT00))
S M=((PWQT10,$ROOT.QT10))
S M=((PWQT20,$ROOT.QT20))
S M=((PWQT50,$ROOT.QT50))
S M=((PWQUPA,$ROOT.QUPA))
S M=((PWQUPF,$ROOT.QUPF))
S M=((PWQU00,$ROOT.QU00))
S M=((PWQU01,$ROOT.QU01))
S M=((PWQU10,$ROOT.QU10))
S M=((PWQU20,$ROOT.QU20))
S M=((PWQV10,$ROOT.QV10))
S M=((PWQV20,$ROOT.QV20))
S M=((PWQV30,$ROOT.QV30))
S M=((PWQX00,$ROOT.QX00))
S M=((PWQX01,$ROOT.QX01))
S M=((PWQY01,$ROOT.QY01))
S M=((PWQY02,$ROOT.QY02))
S M=((PWQY03,$ROOT.QY03))
S M=((PWQY04,$ROOT.QY04))
S M=((PWQY05,$ROOT.QY05))
S M=((PWQY10,$ROOT.QY10))
S M=((PWQY11,$ROOT.QY11))
S M=((PWQY20,$ROOT.QY20))
S M=((PWQY30,$ROOT.QY30))
S M=((PWQZ00,$ROOT.QZ00))
S M=((PWQ000,$ROOT.Q000))
S M=((PWQ100,$ROOT.Q100))
S M=((PWQ101,$ROOT.Q101))
S M=((PWQ102,$ROOT.Q102))
S M=((PWQ103,$ROOT.Q103))
```

```
            S M=((PWQ104,$ROOT.Q104))
            S M=((PWQ200,$ROOT.Q200))
            S M=((PWQ210,$ROOT.Q210))
            S M=((PWQ300,$ROOT.Q300))
            S M=((PWQ400,$ROOT.Q400))
            S M=((PWQ500,$ROOT.Q500))
            S M=((PWQ600,$ROOT.Q600))
            S M=((PWQ700,$ROOT.Q700))
            S M=((PWQ800,$ROOT.Q800))
            S M=((PWQ900,$ROOT.Q900))
            S M=((PWR000,$ROOT.R000))
            S M=((PWR005,$ROOT.R005))
            S M=((PWR100,$ROOT.R100))
            S M=((PWR200,$ROOT.R200))
            S M=((PWR400,$ROOT.R400))
            S M=((PWR500,$ROOT.R500))
            S M=((PWR600,$ROOT.R600))
            S M=((PWR980,$ROOT.R980))
            S M=((PWR990,$ROOT.R990))
   ===SEQ FOR PUF
            S M=((PWFANM,$ROOT.FANM))
            S M=((PWFBIB,$ROOT.FBIB))
            S M=((PWFCTL,$ROOT.FCTL))
            S M=((PWFDBD,$ROOT.FDBD))
            S M=((PWFECR,$ROOT.FECR))
            S M=((PWFENU,$ROOT.FENU))
            S M=((PWFFOG,$ROOT.FFOG))
            S M=((PWFMCL,$ROOT.FMCL))
            S M=((PWFPGM,$ROOT.FPGM))
            S M=((PWFRUB,$ROOT.FRUB))
            S M=((PWFSDO,$ROOT.FSDO))
            S M=((PWFTXT,$ROOT.FTXT))
            S M=((PWFVER,$ROOT.FVER))
            S M=((PWF000,$ROOT.F000))
   ===SEQ FOR SEC
            S M=((PWSECT,$ROOT.SECT))
   ===SEQ
   //*       S M=((PWUCTR,$ROOT.UCTR))
   //*       S M=((PWUCTX,$ROOT.UCTR))
   //*       S M=((PWUCTZ,$ROOT.UCTR))
   //
```

```
//$PRFJ.D4V JOB ($CCPT),'PAC D04VS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              INSTALLATION - D04VS                                 *
//*    INITIAL ALLOCATING AND LOADING OF THE PDS                      *
//*    OF PAF COBOL/VS SUB-PROGRAMS                                   *
//*    (AND RACF SECURITY SYSTEM INTERFACE, IF "SEC" OPTION)          *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*     .STEP3 : LOADING                                              *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($MCVS)
//*
//STEP2  EXEC PGM=IEFBR14
//LNKB     DD DSN=$MCVS,
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6144,(100,10,5)),DCB=(RECFM=U,BLKSIZE=6144)
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTVS    DD DSN=$MCVS,DISP=OLD                BATCH
//INPVS    DD DSN=PAC.MCOBVS,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(20,SL)
//SYSIN    DD *
  C   I=((INPVS,R)),O=OUTVS
          S M=PBBTST
          S M=PBBTWS
          S M=PBBT98
          S M=((PWTPST,$ROOT.TPST))
          S M=((PWTPWS,$ROOT.TPWS))
===SEQ FOR SEC
          S M=PACSECB
===SEQ
/*
//
```

4.4.6. SYSTEM - ALLOC./LOADING OF BATCH PROCEDURES

1.5. ALLOCATION AND LOADING OF BATCH PROCEDURES

It is recommended that all VA Pac operation procedures be cataloged in one procedure library, i.e.:

- Either in a VA Pac-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.

ALLOCATION OF A VA PAC-RESERVED PROCLIB (OPTIONAL)

```
===MOD D05IPROC   Job '$prfj.D5I'

STEP1: IEFBR14:   allocation of the procedure library
```

LOADING OF THE BATCH PROCEDURES IN THE PROCLIB

```
===MOD D05PROC    Job '$prfj.D5P'
```

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 '$radpNNNN', where $radp is the root chosen at installation time, and NNNN the standard IBM name of the procedure.

Procedures are detailed in the Batch Procedures, Administrator's Guide and User's Guide, except for the older-database retrieval procedures, which are described in this part of the manual.

```
//$PRFJ.D5I  JOB ($CCPT),'PAC D05IPROC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              INSTALLATION - D05IPROC                              *
//*    WARNING! OPTIONAL JOB                                          *
//*    ====================                                           *
//*    INITIAL ALLOCATION OF A SPECIAL VA PAC "PROCLIB"               *
//*     .STEP1 : ALLOCATION                                           *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEFBR14
//LIB      DD DSN=$BIBP,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(200,20,10))
```

```
//$PRFJ.D5P JOB ($CCPT),'PAC D05PROC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D05PROC                              *
//*                                                                   *
//*    CATALOGING OF BATCH PROCEDURES                                 *
//*                                                                   *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*                                                                   *
//*********************************************************************
//UPD    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$BIBP,DISP=SHR
//SYSIN    DD DATA,DLM='F+'
:/ ADD NAME=$RADP.ARCH
:/ ADD NAME=$RADP.GPRP
:/ ADD NAME=$RADP.GPRT
:/ ADD NAME=$RADP.GPRU
:/ ADD NAME=$RADP.MLIB
:/ ADD NAME=$RADP.UPDT
:/ ADD NAME=$RADP.UPDP
:/ ADD NAME=$RADP.PACX
:/ ADD NAME=$RADP.PARM
:/ ADD NAME=$RADP.LOAE
:/ ADD NAME=$RADP.REOR
:/ ADD NAME=$RADP.QREO
:/ ADD NAME=$RADP.REST
:/ ADD NAME=$RADP.RESY
:/ ADD NAME=$RADP.SAVE
:/ ADD NAME=$RADP.SASY
:/ ADD NAME=$RADP.REAG
:/ ADD NAME=$RADP.SVAG
:/ ADD NAME=$RADP.VINS
:/ ADD NAME=$RADP.PPAF
:/ ADD NAME=$RADP.SADM
:/ ADD NAME=$RADP.YSMC
:/ ADD NAME=$RADP.STOP
:/ ADD NAME=$RADP.EMLD
:/ ADD NAME=$RADP.EMUP
===SEQ FOR TAB
:/ ADD NAME=$RADP.GETD
:/ ADD NAME=$RADP.GETA
:/ ADD NAME=$RADP.GETI
:/ ADD NAME=$RADP.SMTD
:/ ADD NAME=$RADP.RMTD
:/ ADD NAME=$RADP.GET0
:/ ADD NAME=$RADP.GET1
:/ ADD NAME=$RADP.GET2
===SEQ
:/ ADD NAME=$RADP.CPSN
:/ ADD NAME=$RADP.EMSN
:/ ADD NAME=$RADP.MESN
:/ ADD NAME=$RADP.SASN
:/ ADD NAME=$RADP.ACTI
:/ ADD NAME=$RADP.UXSR
:/ ADD NAME=$RADP.PQCA
:/ ADD NAME=$RADP.PQCE
:/ ADD NAME=$RADP.RVDE
:/ ADD NAME=$RADP.RVKE
===SEQ FOR PEI
:/ ADD NAME=$RADP.INPE
:/ ADD NAME=$RADP.SVPE
:/ ADD NAME=$RADP.RSPE
:/ ADD NAME=$RADP.PRPE
:/ ADD NAME=$RADP.GRPE
:/ ADD NAME=$RADP.HIPE
:/ ADD NAME=$RADP.SIPE
===SEQ FOR VISP
:/ ADD NAME=$RADP.VDWN
:/ ADD NAME=$RADP.VPUR
:/ ADD NAME=$RADP.VUP1
:/ ADD NAME=$RADP.VUP2
===SEQ FOR PACTR
:/ ADD NAME=$RADP.TRDU
```

```
    :/ ADD NAME=$RADP.TRJC
    :/ ADD NAME=$RADP.TRPF
    :/ ADD NAME=$RADP.TRRP
    :/ ADD NAME=$RADP.TRRT
    :/ ADD NAME=$RADP.TRUP
    ===SEQ FOR S2K
    :/ ADD NAME=$RADP.IANA
    :/ ADD NAME=$RADP.IGRA
    :/ ADD NAME=$RADP.INFP
    :/ ADD NAME=$RADP.INFQ
    :/ ADD NAME=$RADP.IPEP
    :/ ADD NAME=$RADP.IPFQ
    :/ ADD NAME=$RADP.IPIA
    :/ ADD NAME=$RADP.ISEP
    ===SEQ
    :/ ADD NAME=$RADP.PRGS
    :/ ADD NAME=$RADP.XPAF
    :/ ADD NAME=$RADP.XPDM
    :/ ADD NAME=$RADP.INSL
    :/ ADD NAME=$RADP.RTLO
    :/ ADD NAME=$RADP.CRYP
    :/ ADD NAME=$RADP.LVBL
    :/ ADD NAME=$RADP.CSES
    :/ ADD NAME=$RADP.ESES
    ===SEQ FOR ROLD
    :/ ADD NAME=$RADP.PJ16
    :/ ADD NAME=$RADP.PP16
    :/ ADD NAME=$RADP.RPPG
    ===SEQ FOR TEAM
    :/ ADD NAME=$RADP.TCCI
    :/ ADD NAME=$RADP.TCGP
    :/ ADD NAME=$RADP.TCLS
    :/ ADD NAME=$RADP.TCME
    ===SEQ
    F+
    //
```

4.4.7. SYSTEM - LOADING OF GENERATION SKELETON FILES

1.6 LOADING OF GENERATION 'SKELETON' FILES

```
===MOD D06SKEL     Job '&prfj.D6'

STEP1: IDCAMS:     DELETE/DEFINE of files SC, SG, SR, SS,
                   and SP

STEP2: IDCAMS:     Loading (REPRO) of SC, SG, SR, SS,
                   and SP

STEP3: IDCAMS:     DELETE of SF file;

STEP4: IEBGENER:   Loading of SF file.
```

```
//$PRFJ.D6  JOB ($CCPT),'PAC D06SKEL',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D06SKEL                              *
//*    LOADING OF VA PAC SKELETON FILES FOR GENERATION                *
//*     .STEP1 : DELETE DEFINE                                        *
//*     .STEP2 : LOADING VSAM SKELETON FILES                          *
//*     .STEP3 : DELETE "SF"                                          *
//*     .STEP4 : ALLOCATING AND LOADING "SF"                          *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SC),DISP=SHR
//         DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SG),DISP=SHR
//         DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SR),DISP=SHR
//         DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SS),DISP=SHR
//         DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.SP),DISP=SHR
//*
//STEP2  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SCO      DD DSN=$INDSV..$ROOT.$ROOT.SC,DISP=SHR
//SGO      DD DSN=$INDSV..$ROOT.$ROOT.SG,DISP=SHR
//SRO      DD DSN=$INDSV..$ROOT.$ROOT.SR,DISP=SHR
//SSO      DD DSN=$INDSV..$ROOT.$ROOT.SS,DISP=SHR
//SPO      DD DSN=$INDSV..$ROOT.$ROOT.SP,DISP=SHR
//SCI      DD DSN=PAC.SCB,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(24,SL),
//         DCB=(RECFM=FB,LRECL=3204,BLKSIZE=3204)
//SGI      DD DSN=PAC.SGB,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(25,SL),
//         DCB=(RECFM=FB,LRECL=4605,BLKSIZE=4605)
//SRI      DD DSN=PAC.SRB,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(26,SL),
//         DCB=(RECFM=FB,LRECL=4605,BLKSIZE=4605)
//SSI      DD DSN=PAC.SSB,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(27,SL),
//         DCB=(RECFM=FB,LRECL=4605,BLKSIZE=4605)
//SPI      DD DSN=PAC.SP,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(13,SL),
//         DCB=(RECFM=FB,LRECL=4605,BLKSIZE=4605)
//SYSIN    DD *
  REPRO INFILE (SCI)   OUTFILE (SCO)
  REPRO INFILE (SGI)   OUTFILE (SGO)
  REPRO INFILE (SRI)   OUTFILE (SRO)
  REPRO INFILE (SSI)   OUTFILE (SSO)
  REPRO INFILE (SPI)   OUTFILE (SPO)
//*
//STEP3  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOT.$ROOT.SF)
 DELETE ($INDSN..$ROOT.$ROOT.SF1)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=PAC.SF,DISP=SHR,UNIT=$U3480,
//         VOL=(,RETAIN,SER=$ICART),LABEL=(14,SL),
//         DCB=(RECFM=FB,LRECL=119,BLKSIZE=11900)
//SYSUT2   DD DSN=$INDSN..$ROOT.$ROOT.SF,DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,(10,1)),
//         DCB=(RECFM=FB,LRECL=119,BLKSIZE=11900)
//STEP5  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=PAC.SFCOBVS,DISP=SHR,UNIT=$U3480,
//         VOL=(,RETAIN,SER=$ICART),LABEL=(21,SL),
//         DCB=(RECFM=FB,LRECL=119,BLKSIZE=11900)
//SYSUT2   DD DSN=$INDSN..$ROOT.$ROOT.SF1,DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
```

```
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,(10,1)),
//         DCB=(RECFM=FB,LRECL=119,BLKSIZE=11900)
//*
//
```

4.4.8. SYSTEM - LOADING OF ERR.MESS, DOC, USER PARAM.

1.7. LOADING OF ERROR MESSAGES, DOCUMENTATION AND USER PARAMETERS

```
===MOD D07AE0     Job '$prfj.D7'
```

This job loads the sequential source file (AE0) on disk and initializes the indexed system file (AE) using AE0, if AE does not already exist at the site.

```
STEP1: IDCAMS:    DELETE of AE0.

STEP2: IEBGENER:  Allocation and loading of AE0.

STEP3: IDCAMS:    Validation of existence of AE (LISTCAT).

STEP4: IDCAMS:    DELETE/DEFINE of AE if AE does not exist.

STEP5: IDCAMS:    REPRO of AE0 to AE if AE does not exist.


===MOD D07PARM    Job '&prfj.D7P'
```

This loading phase performs the installation of user parameters in the system AE and AP files.

STEP1: '$radpPARM': execution of the PARM procedure.

WARNING:

THE OPERATION OF THE VA PAC SYSTEM REQUIRES AT LEAST ONE OF THE FOLLOWING USER PARAMETERS:

- USER CODE
- VA PAC SYSTEM ACCESS KEY

Default-supplied input include:

- One NU line: test user code providing all access authorizations to the databases.

  Code: TEST
  Password: none

This code is used to perform installation tests; it must be deleted once the user code of the VA Pac database administrator is entered.

- One NK line: this must be filled with the VA Pac ACCESS KEY SPECIFIED IN THE INSTALLATION MAILING.

- NC lines, examples of optional control cards for generation, and other NU lines, examples of User codes.

WORKSTATION/PACDESIGN - METHODOLOGY PARAMETERIZATION

Enter the parameters corresponding to the methodology or methodologies in use.

Update transactions are found in the Methodology PDS &INDSN..&ROOT.&ROOT.SYMT.

```
. Merise.........................: PARMMER

. YSM (Yourdon Structured Method)...: PARMYSM

. SSADM..........................: PARMADM

. IFW............................: PARMIFW

. OMT............................: PARMOMT
```

Copy the member(s) corresponding to the site's methodology or methodologies in the SYSIN.

```
//$PRFJ.D7  JOB ($CCPT),'PAC D07AE0',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D07AE0                               *
//*    INITIAL LOADING OF VA PAC'S AE0 AND AE SYSTEM FILES            *
//*     STEP1 : DELETE "AE0"                                          *
//*     STEP2 : ALLOCATING AND LOADING "AE0"                          *
//*     STEP3 : CHECKING THE EXISTENCE OF "AE" FILE                   *
//*     STEP4 : DELETE DEFINE "AE" IF IT DOESN'T EXIST                *
//*     STEP5 : INITIAL LOADING "AE" IF IT DOESN'T EXIST              *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOT.$ROOT.AE0)
//*
//STEP2  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=PAC.AE0A,DISP=SHR,UNIT=$U3480,
//         VOL=(,RETAIN,SER=$ICART),LABEL=(23,SL),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440)
//SYSUT2   DD DSN=$INDSN..$ROOT.$ROOT.AE0,DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,(120,15)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560)
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDSN..$ROOT.$ROOT.SY(LI$ROOT.$ROOT.AE),
//         DISP=SHR
//*
//STEP4  EXEC PGM=IDCAMS,COND=(0,EQ,STEP3)
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.AE),DISP=SHR
//*
//STEP5  EXEC PGM=IDCAMS,COND=(0,EQ,STEP3)
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//AEO      DD DSN=$INDSV..$ROOT.$ROOT.AE,DISP=SHR
//AEI      DD DSN=$INDSN..$ROOT.$ROOT.AE0,DISP=OLD
//SYSIN    DD *
    REPRO INFILE (AEI) OUTFILE (AEO)
//
```

```
===FRM TYPE=DATA
//$PRFJ.D7P JOB ($CCPT),'PAC D07PARM',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D07PARM                              *
//*    LOADING OF USER PARAMETERS INTO AE AND AP SYSTEM FILES.        *
//*********************************************************************
//* VERY IMPORTANT - VERY IMPORTANT - VERY IMPORTANT - VERY IMPORTANT *
//* -------------- - -------------- - -------------- - ---- --------- *
//*                                                                   *
//* 1) DON'T FORGET TO INDICATE ON A ' NK' LINE                       *
//*    THE PACBASE ACCESS KEY GIVEN BY VA PACBASE SUPPORT.            *
//*                                                                   *
//* 2) UTILIZATION OF PACDESIGN - METHOD UPDATING                     *
//*    ADD INTO THE "SYSIN" THE TRANSACTIONS CORRESPONDING            *
//*    TO THE METHODS THAT ARE USED WITH THE WORKSTATION INSTALLATION.*
//*    THOSE BATCH TRANSACTIONS CAN BE FOUND IN THE PDS FILE          *
//*    $INDSN..&ROOT.&ROOT.SYMT                                       *
//*    .YSM (YOURDON STRUCTURED METHOD) ....... : PARMYSM             *
//*    .SSADM ................................. : PARMADM             *
//*    .MERISE ................................ : PARMMER             *
//*    .IFW ................................... : PARMIFW             *
//*    .OMT ................................... : PARMOMT             *
//*    COPY IN THE SYSIN BELOW THE CORRESPONDING MEMBER(S) "PARM...". *
//*                                                                   *
//*********************************************************************
//STEP1  EXEC $RADP.PARM
//SYSIN    DD *
 *********
 NRCHAR <--- REQUIRED LOADING COMMAND AT TIME OF INSTALLATION
 NK000<---------------    ACCESS KEY    ------------------------->
 NCAE  EXAMPLE OF COBOL PROGRAM COMPILE
 NCAE01//$PRFJ.COB JOB ($CCPT),'COMPIL',CLASS=$CLASSJ,
 NCAE02//     MSGCLASS=$MSGCL
 NCAE03//*** EXAMPLE OF COBOL PROGRAMS COMPILE
 NCAE04//*** CORRECT THE FOLLOWING COBOL COMPILE LINES
 NCAE05//   EXEC COBVCL,S=X,BIBLI='............'
 NCAE06//COB.SYSIN DD *
 NCDE  CBL NAME LINE
 NCDE01 CBL NAME
 NCZI  EXAMPLE OF END OF JOB STREAM LINES: LINK-EDIT
 NCZI01//*  END OF JOB STREAM LINES: PUT LINK LIBRARY
 NCZI02//LKED.SYSLIB  DD
 NCZI03//             DD DSN=.............,DISP=SHR
 NCZI04//*********************************************************
 NCZI05//
 NCAA  ANOTHER EXAMPLE WITH PARAMETERS
 NCAA01//- JOB (-),-,CLASS=-,MSGCLASS=-                          12U34
 NCAA02//*  ANOTHER EXAMPLE OF THE BEGINNING-OF-JOB-STREAM LINE
 NCAA03//*  WITH PARAMETERS MARKED BY DASHES ('-') ON THE LINE, AND
 NCAA04//*  OF COMMANDS INDICATING THE NATURE OF THE PARAMETERS IN
 NCAA05//*  COLUMNN 75 (P = NAME OF GENERATED PROGRAM, U = USER
 NCAA06//*  CODE, NUMBERS 1 TO 9 = PARAMETERS DECLARED IN THE
 NCAA07//*  PACBASE DATABASE, ETC.)
 NCDA  ./ ADD NAME=PROGRAM NAME LINE
 NCDA01./ ADD NAME=-                                             P
 NCFA  /* LINE
 NCFA01/*
 NCZA01//
 NCZA  // LINE
 NUXXXXXXXX000PPPPPPPP 1
 NUXXXXXXXX010$ROOT.$FILE
 NUTEST    000        44
 NUTEST    010    $E 3                                           4
/*
//
```

4.4.9. SYSTEM - LIST OF PROGRAMS AND INSTALL. FILES

1.End. LIST OF PROGRAMS AND INSTALLATION FILES
```
===MOD D99INSL    Job '$prfj.D9'
```

This job prints the following lists:

- A list of the batch programs, the maps, and the on-line programs, with their generation characteristics.
- A list of permanent system files AE, SC, SG, SR, SS, SP, SI, and SF, with their building dates.

These listings should be kept to be sent to VisualAge Pacbase Support., together with the installation references, in case of system operation problems.

This job also executes the INSL procedure.

User input:

Pos.	Len.	Value	Meaning
3	2	rr	rr=ROOT PACBASE system root

Meaning of return codes:

```
4:    Insufficient storage (increase the value of the
                            REGION parameter)

6:    Load-module not found in the library

A:    Not a standard VA Pac program
```

Contact VisualAge Pacbase Support if the return code has one of the following values:

```
5:    Unknown error

7:    Input/output error

8:    System error

9:    Loading error (system error)
```

```
//$PRFJ.D99  JOB ($CCPT),'PAC D99INSL',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - D99INSL                              *
//*    THE FOLLOWING JOB PROVIDES THE LIST OF PROGRAMS                *
//*    AND SYSTEM FILES INSTALLED ON THE SITE.                        *
//*********************************************************************
//INSL   EXEC $RADP.INSL
  $ROOT
//
```

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*             - LIST OF INSTALLED PROGRAMS AND FILES -              *
//*********************************************************************
//$RADP.INSL  PROC ROOT=$ROOT,   ROOT OF THE PACBASE SYSTEM
//         INDSV='$INDSV',      INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',      INDEX OF SYSTEM NON VSAM FILES
//*:       SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//         STEPLIB='$MODB',   LIBRARY OF LOAD-MODULES
//         MODT='$MODT',      LIBRARY OF INSTALLED TP LOAD MODULES
//         OUT=$OUT,            OUTPUT CLASS
//         UWK=$UWK             WORK FILES UNIT
//*********************************************************************
//*   INPUT:                                                          *
//*   - ONE LINE: ROOT &ROOT OF THE VA PAC SYSTEM (COL.3 LENGTH 2)    *
//*********************************************************************
//INPUT  EXEC PGM=PTU001
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&INSLMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7SC   DD DSN=&INDSV..&ROOT.&ROOT.SC,DISP=SHR
//PAC7SG   DD DSN=&INDSV..&ROOT.&ROOT.SG,DISP=SHR
//PAC7SR   DD DSN=&INDSV..&ROOT.&ROOT.SR,DISP=SHR
//PAC7SP   DD DSN=&INDSV..&ROOT.&ROOT.SP,DISP=SHR
//PAC7SS   DD DSN=&INDSV..&ROOT.&ROOT.SS,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFSC),DISP=SHR
//         DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFSG),DISP=SHR
//         DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFSR),DISP=SHR
//         DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFSP),DISP=SHR
//         DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFSS),DISP=SHR
//PACDTP EXEC PGM=PACDTP
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=&MODT,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7SC   DD DSN=&INDSV..&ROOT.&ROOT.SC,DISP=SHR
//PAC7SG   DD DSN=&INDSV..&ROOT.&ROOT.SG,DISP=SHR
//PAC7SR   DD DSN=&INDSV..&ROOT.&ROOT.SR,DISP=SHR
//PAC7SS   DD DSN=&INDSV..&ROOT.&ROOT.SS,DISP=SHR
//PAC7SP   DD DSN=&INDSV..&ROOT.&ROOT.SP,DISP=SHR
//PAC7SF   DD DSN=&INDSN..&ROOT.&ROOT.SF,DISP=SHR
//PAC7DS   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&INSLMB,DISP=(OLD,PASS)
//SYSUDUMP DD SYSOUT=&OUT
```

4.4.10. SYSTEM COMPLEMENT - OLSD MULTI-SCREEN VARIANT

2. OPTIONAL COMPLEMENTS OF THE VA PAC SYSTEM INSTALLATION

2.1. ALLOCATION & LOADING OF THE OLSD MULTI-SCREEN VARIANT SOURCES

This installation complement is necessary only for the users of the multi-screen variant of the OLSD function.

It loads the &INDSN..&ROOT.&ROOT.MSO PDS

```
===MOD E01MSO     Job '$prfj.E1M'

STEP1: IDCAMS:    DELETE of the PDS.
STEP2: IEFBR14:   allocation of the PDS.
STEP3: IEBCOPY:   loading of the PDS members.
```

```
//$PRFJ.E1M  JOB ($CCPT),'PAC E01MSO',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              INSTALLATION - E01MSO                                *
//*   OPTIONAL: OLSD "MULTI-SCREEN" OPTION                            *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*     .STEP3 : LOADING                                              *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*   THIS PDS CONTAINS DISPLAY SUB-PROGRAMS                          *
//*   OF ALL OLSD MULTI-SCREEN VARIANTS.                              *
//*   THE USER CAN KEEP AND ADAPT SOURCE SUB-PROGRAMS NECESSARY       *
//*   FOR THE OLSD VARIANT(S) OF THE SITE.                            *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOT.$ROOT.MSO)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDSN..$ROOT.$ROOT.MSO,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6080,(250,25,10))
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTM     DD DSN=$INDSN..$ROOT.$ROOT.MSO,DISP=OLD
//INM      DD DSN=PAC.SOURCE,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(19,SL)
//SYSIN    DD *
  C   I=INM,O=OUTM
  S M=HPFORM
  S M=PACVMSS
  S M=SCRCODIF
  S M=SCRIOPAR
  S M=SCRPEINT
  S M=SCRSAISI
  S M=SCRDEC
  S M=SCRMFO
  S M=VMSUTIL
  S M=ZARBUR
  S M=ZARCVS
  S M=ZARDEC
  S M=ZARDE2
  S M=ZARG7
  S M=ZARG8
  S M=ZARICL
  S M=ZARIMS
  S M=ZARMFO
  S M=ZARMF1
  S M=ZARTRM
//
```

4.4.11. SYSTEM COMPLEMENT - DB2 ACCESS SQL SOURCE PGMS

2.2 ALLOCATION & LOADING OF SOURCE PROGRAMS FOR ACCESS TO SQL OPTION'S DB2 CATALOG

This installation complement is necessary only for those users of the SQL variant.

It loads the &INDSN..&ROOT.&ROOT.SQL PDS.

```
===MOD E02SQL      Job '&prfj.E2S'
```

STEP1: IDCAMS: DELETE of the PDS
STEP2: IEFBR14: Allocation of the PDS
STEP3: IEBCOPY: Loading of PDS members

```
===FRM TYPE=DATA
===SEQ FOR SQL
//$PRFJ.E1Q JOB ($CCPT),'VSAM DEF.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - E01SQL                               *
//*    OPTIONAL: ALLOCATION AND LOADING OF SQL COBOL SOURCE PROGRAMS  *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOT.$ROOT.SQL)
//*
//STEP2  EXEC PGM=IEFBR14
//SQL      DD DSN=$INDSN..$ROOT.$ROOT.SQL,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6080,(25,2,5))
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTSQL   DD DSN=$INDSN..$ROOT.$ROOT.SQL,DISP=SHR
//INPSQL   DD DSN=PAC.SRCSQL,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(18,SL)
//SYSIN    DD *
  C I=((INPSQL,R)),O=OUTSQL
//
===SEQ
```

4.4.12. SYSTEM COMPLEMENT - SECURITY SYSTEM INTERFACE

2.3. ALLOCATION & LOADING OF THE SECURITY SYSTEM INTERFACE SUB-PROGRAMS (SEC VARIANT)

This installation complement is necessary only for users of the SEC variant. (For further details, see the SECURITY SYSTEM INTERFACE Reference manual).

RACF

```
===MOD E02SECR    Job '$prfj.E2R'

STEP1: IEBCOPY:   loading of the PACSECU8 RACF Table access
                  sub-program in an AUTHORIZED LIBRARY.
```

- Definition of the resource class (macros ICHERCDE and ICHRFRTB), for instance:

```
cccc     ICHERCDE CLASS=cccc,                                  *
               ID=128,                                         *
               MAXLNTH=4,                                      *
               FIRST=ALPHANUM,                                 *
               OTHER=ANY,                                      *
               POSIT=25,                                       *
               OPER=NO

cccc     ICHRFRTB CLASS=cccc,ACTION=RACF
```

cccc being the RACF resource class code reserved for VA Pac (see PARM procedure, User input, line code 'NS').

- Declaration of PERMITs for the VA Pac resource declaration option under RACF, for instance:

```
PERMIT 3£££ CLASS(cccc) ID(user) ACCESS(ALTER)
PERMIT 3$P CLASS(cccc) ID(user) ACCESS(ALTER)
PERMIT 3$E CLASS(cccc) ID(user) ACCESS(ALTER)
PERMIT 3BBB CLASS(cccc) ID(user) ACCESS(ALTER)

etc.
```

TOPSECRET

```
===MOD E02SECT    Job '$prfj.E2T'

STEP1: IDCAMS:    DELETE of the sub-program source library
STEP2: IEFBR14:   allocation of the library
STEP3: IEBCOPY:   loading of the sources of the sub-programs
                  (assembler) for access to the TOPSECRET
                  Tables to be compiled.
```

- Compilation of the access sub-programs.

For the compilation of PACTSS and PACTSSC, the OPMAT TSS library must be specified in the SYSLIB card of the Assembler compilation program.

PACTSSC is a CICS program. It must be translated before its compilation and link-edit.

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

- Definition of the resource class:

TSS ADD (RDT) RESCLASS(cccc) RESCODE(xx)

```
cccc = code of the RACF resource class corresponding to
       VA Pac (see the NS line code of the User input of
       the PARM procedures).
xx = hexadecimal code indicating the type of resource.
```

- Resource creation

TSS ADD(dept_name) cccc(nlib) cccc(nlib) ...

dept_name: department name
n: authorization level
lib= VA Pac library code

- Definition of access authorizations

TSS PERMIT(user_code) cccc(nlib) TSS PERMIT(user_code) cccc(nlib)

```
===SEQ FOR SEC
//$PRFJ.E2R JOB ($CCPT),'PAC E02SECR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - E02SECR RACF                         *
//*    LOADING OF THE RACF SECURITY SYSTEM INTERFACE MODULE           *
//*     .STEP1 : LOADING                                              *
//*********************************************************************
//STEP1    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OULIB    DD DSN=-----,DISP=OLD         <-- AUTHORIZED LIBRARY
//INLIB    DD DSN=PAC.MBR8,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(4,SL)
//SYSIN    DD *
  C I=((INLIB,R)),O=OULIB
  S M=((PACSECRA,PACSECU8))
/*
//
===SEQ
```

```
===SEQ FOR SEC
//$PRFJ.E2T JOB ($CCPT),'PAC E02SECT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - E02SECT TOPSECRET                    *
//*    LOADING OF THE TOPSECRET SECURITY SYSTEM INTERFACE MODULE      *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*     .STEP3 : LOADING                                              *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*   THIS PDS CONTAINS THE TOPSECRET TABLES ACCESS SUB-PROGRAMS      *
//*   WHICH MUST BE COMPILED ON THE SITE.                             *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOT.$ROOT.TSS)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDSN..$ROOT.$ROOT.TSS,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6080,(250,25,10))
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTM     DD DSN=$INDSN..$ROOT.$ROOT.TSS,DISP=OLD
//INM      DD DSN=PAC.SOURCE,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(19,SL)
//SYSIN    DD *
  C   I=INM,O=OUTM
  S M=PACTSS
  S M=PACTSSC
//
===SEQ
```

4.4.13. DATABASE - ALLOCATION OF PARAMETER PDS

3. INSTALLATION OF THE VA PAC DATABASE

3.1 INITIAL ALLOCATION OF THE PARAMETER PDS

Database parameters: &INDUN...&ROOT.&FILE.SY

```
===MOD I01SY      Job '&prfj.Il'

STEP1: IDCAMS:    DELETE of the PDS

STEP2: IEFBR14:   allocation
```

```
//$PRFJ.I1   JOB ($CCPT),'PAC I01SY',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I01SY                                *
//* INITIAL ALLOCATION OF THE PARAMETERS PDS OF ONE VA PAC DATABASE   *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDUN..$ROOT.$FILE.SY)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDUN..$ROOT.$FILE.SY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(6080,(100,,10))
//
```

4.4.14. DATABASE - LOADING OF PARAMETER PDS

3.2 LOADING OF THE PARAMETER PDS

This PDS contains the definitions of the database files. Specified file sizes relate to the installation test database. They should therefore be user-modified according to the evolution of the database size.

CAUTION: be sure to replace :/ with ./ before submitting the job.

```
===MOD I02SY      Job '$prfj.I2'

STEP1: IEBUPDTE:  Loading
```

```
===FRM TYPE=DATA
//$PRFJ.I2   JOB ($CCPT),'PAC I02SY',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I02SY                                *
//*    LOADING OF THE PARAMETER PDS                                   *
//*     .STEP1 : LOADING PARAMETERS FOR ONE VA PAC DATABASE           *
//*    ->NOTE 1                                                       *
//*      ------                                                       *
//*      REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                   *
//*    ->NOTE 2                                                       *
//*      ------                                                       *
//*      THIS JOB CONTAINS THE SYSIN'S FOR ALLOCATING THE FILES       *
//*      THAT MAKING UP THE VA PAC DATA BASE:                         *
//*      ADAPT THE SIZES SPECIFIED IN THE 'DEFINE' STATEMENTS         *
//*      TO YOUR REQUIREMENTS.                                        *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDUN..$ROOT.$FILE.SY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/     ADD NAME=DF$ROOT.$FILE.AN
 DELETE ($INDUV..$ROOT.$FILE.AN) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.AN)   -
                  SHR (2)     RUS   KEYS (43,0)   -
                  VOL ($VOLUV) -
                  CYL (4 1)    -
                  RECSZ (54,54) )                 -
        INDEX   ( NAME ($INDUV..$ROOT.$FILE.AN.I) -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDUV..$ROOT.$FILE.AN.D) -
                  FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOT.$FILE.AR
 DELETE ($INDUV..$ROOT.$FILE.AR) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.AR)   -
                  SHR (2)    RUS   NUMBERED       -
                  VOL ($VOLUV) -
                  CYL (4 1)    -
                  RECSZ (140,140)  )              -
       DATA    ( NAME ($INDUV..$ROOT.$FILE.AR.D)  -
       CISZ (2048) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOT.$FILE.AJ
 DELETE ($INDUV..$ROOT.$FILE.AJ) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.AJ)   -
                  SHR (2)    RUS   NUMBERED       -
                  VOL ($VOLUV) -
                  CYL (1 1)    -
                  RECSZ (167,167)  )              -
        DATA    ( NAME ($INDUV..$ROOT.$FILE.AJ.D) -
        CISZ (512) )          /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOT.$FILE.AG
 DELETE ($INDUV..$ROOT.$FILE.AG) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.AG)   -
                  SHR (2)     RUS   KEYS (27,0)   -
                  VOL ($VOLUV) -
                  CYL (1 1)    -
                  RECSZ (150,150) )               -
        INDEX   ( NAME ($INDUV..$ROOT.$FILE.AG.I) -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDUV..$ROOT.$FILE.AG.D) -
                  FSPC (50,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
===SEQ FOR PEI
:/     ADD NAME=DF$ROOT.$FILE.AB
 DELETE ($INDUV..$ROOT.$FILE.AB) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.AB)   -
                  SHR (4)   RUS   KEYS (26,0)     -
                  VOL ($VOLUV) -
                  CYL (1 1)    -
                  RECSZ (110,110) )               -
        INDEX   ( NAME ($INDUV..$ROOT.$FILE.AB.I) -
                  CISZ (1024) )                   -
        DATA    ( NAME ($INDUV..$ROOT.$FILE.AB.D) -
```

```
                  FSPC (10,5)                    -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
    :/     ADD NAME=DF$ROOT.$FILE.AC
     DELETE ($INDUV..$ROOT.$FILE.AC) CLUSTER
     DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.AC)   -
                  SHR (2 3)   RUS   KEYS (26,0)   -
                  VOL ($VOLUV) -
                  CYL (1 1)    -
                  RECSZ (110,110) )              -
            INDEX   ( NAME ($INDUV..$ROOT.$FILE.AC.I) -
                  CISZ (1024) )                  -
            DATA    ( NAME ($INDUV..$ROOT.$FILE.AC.D) -
                  FSPC (10,5)                    -
            CISZ (4096) )         /*: CATALOG ($VCAT) */
    ===SEQ FOR PACTR
    :/     ADD NAME=DF$ROOT.$FILE.UV
     DELETE ($INDUV..$ROOT.$FILE.UV) CLUSTER
     IF LASTCC LE 8 THEN DO
     DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.UV) -
                  SHR (2)    RUS  KEYS (19,1)     -
                  VOL ($VOLUV)                    -
                  CYL (5,1)                       -
                  RECSZ (80 80) )                 -
          INDEX   ( NAME ($INDUV..$ROOT.$FILE.UV.I) -
                  CISZ (4096) )                   -
          DATA    ( NAME ($INDUV..$ROOT.$FILE.UV.D) -
                  FSPC (10,5)                     -
          CISZ (4096) )   /*: CATALOG ($SCAT) */
     END
    :/     ADD NAME=DL$ROOT.$FILE.JT
     DELETE ($INDUN..$ROOT.$FILE.JT)
    :/     ADD NAME=DL$ROOT.$FILE.TJ
     DELETE ($INDUN..$ROOT.$FILE.TJ)
    ===SEQ FOR S2K
    :/     ADD NAME=DF$ROOT.$FILE.FP
     DELETE ($INDUV..&USER$ROOT.$FILE.FP) CLUSTER
     DEFINE CLUSTER ( NAME ($INDUV..&USER$ROOT.$FILE.FP)   -
                  SHR (2)     RUS   KEYS (9,0)    -
                  VOL ($VOLUV) -
                  CYL (1 1)    -
                  RECSZ (9,9) )                   -
            INDEX   ( NAME ($INDUV..&USER$ROOT.$FILE.FP.I) -
                  CISZ (4096) )                   -
            DATA    ( NAME ($INDUV..&USER$ROOT.$FILE.FP.D) -
                  FSPC (10,5)                     -
            CISZ (4096) )         /*: CATALOG ($VCAT) */
    ===SEQ FOR VISP
    :/ ADD NAME=DF$ROOT.$FILE.VP
     DELETE ($INDUV..$ROOT.$FILE.VP) CLUSTER
     IF LASTCC LE 8 THEN DO
     DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.VP)   -
                  SHR (2)     RUS  KEYS (11,0)   -
                  CYL (5 1)    -
                  RECSZ (80,80) )                -
            INDEX   ( NAME ($INDUV..$ROOT.$FILE.VP.I) -
                  CISZ (4096) )                   -
            DATA    ( NAME ($INDUV..$ROOT.$FILE.VP.D) -
                  FSPC (10,5)                     -
            CISZ (4096) )
     END
    :/ ADD NAME=DL$ROOT.$FILE.VP
     DELETE ($INDUV..$ROOT.$FILE.VP) CLUSTER
    ===SEQ
    :/     ADD NAME=DF$ROOT.$FILE.EM
     DELETE ($INDUV..$ROOT.$FILE.EM) CLUSTER
     DEFINE CLUSTER ( NAME ($INDUV..$ROOT.$FILE.EM)   -
                  SHR (2 3)   RUS   KEYS (17,0)   -
                  VOL ($VOLUV) -
                  CYL (1 1)    -
                  RECSZ (90,90) )                 -
            INDEX   ( NAME ($INDUV..$ROOT.$FILE.EM.I) -
                  CISZ (1024) )                   -
            DATA    ( NAME ($INDUV..$ROOT.$FILE.EM.D) -
                  FSPC (10,5)                     -
            CISZ (4096) )         /*: CATALOG ($VCAT) */
    :/     ADD NAME=LI$ROOT.$FILE.AJ
     LISTCAT ENTRIES ($INDUV..$ROOT.$FILE.AJ) CLUSTER
```

```
PP
//
```

4.4.15. DATABASE - DATABASE FILE INITIALIZATIONS

3.3 INITIALIZATION OF THE DATABASE FILES

These jobs must be executed every time a VA Pac database is initialized.

- Definition of the database files

```
===MOD I03DEF     Job '$prfj.I3'

STEP1: IDCAMS:    DELETE/DEFINE
```

- GDG allocation - Initilization of files

WARNING: If SMS is present at the site, delete the DD //GDGMOD cards from the IDCAMS steps before submitting the jobs.

```
===MOD I03INI     Job '$prfj.I3I'

STEP1: IDCAMS:    GDG of the PJ file
STEP2: IEBGENER:  Initialization of the PJ file
STEP3: IDCAMS:    GDG of PC file
STEP4: IEBGENER:  Initialization of the PC file (test dbase)
STEP5: IDCAMS:    GDG of PG file
STEP6: IDCAMS:    GDG of PP file (PEI function)
```

```
//$PRFJ.I3  JOB ($CCPT),'PAC I03DEF',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I03DEF                               *
//*    DEFINITION OF THE VSAM FILES FOR ONE VA PAC DATABASE           *
//*     STEP1 : DELETE/DEFINE OF THE FILES                            *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.AN),DISP=SHR
//      DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.AR),DISP=SHR
//      DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.AJ),DISP=SHR
//      DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.AG),DISP=SHR
===SEQ FOR PEI
//      DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.AB),DISP=SHR
//      DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.AC),DISP=SHR
===SEQ
//*
//
```

```
//$PRFJ.I3I JOB ($CCPT),'PAC I03INI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I03INI                               *
//*   JOB TO RUN ONLY FOR THE INITIALIZATION OF A NEW DATABASE        *
//*    . BUILDING OF INDEX DATA-GROUP FOR                             *
//*      "PC", "PJ", "PG" ("PP" IF PEI) SAVE FILES                    *
//*    . "PJ" FILE INITIALIZATION                                     *
//*    . LOADING OF TEST DATABASE ON "PC" FILE                        *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS            *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDUN..$ROOT.$FILE.PJ,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=FB,LRECL=167,BLKSIZE=6179)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$ROOT.$FILE.PJ) LIMIT (3) SCR)
//*
//STEP2  EXEC PGM=IEBGENER
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=167,BLKSIZE=167)
//SYSUT2   DD DSN=$INDUN..$ROOT.$FILE.PJ(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,1),
//         DCB=($DSCB,RECFM=FB,LRECL=167,BLKSIZE=6179)
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD1  DD DSN=$INDUN..$ROOT.$FILE.PC,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=VB,LRECL=155,BLKSIZE=27280)
//GDGMOD2  DD DSN=$INDUN..$ROOT.$FILE.PD,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=VB,LRECL=155,BLKSIZE=27280)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$ROOT.$FILE.PC) LIMIT (3) SCR)
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$ROOT.$FILE.PD) LIMIT (3) SCR)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=PAC.PCB,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(28,SL)
//SYSUT2   DD DSN=$INDUN..$ROOT.$FILE.PC(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,(220,10),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=155,BLKSIZE=27280)
//*
//STEP4B EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=PAC.PCB,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(28,SL)
```

```
//SYSUT2   DD DSN=$INDUN..$ROOT.$FILE.PD(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,(220,10),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=155,BLKSIZE=27280)
//*
//STEP5  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDUN..$ROOT.$FILE.PG,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=FB,LRECL=150,BLKSIZE=6150)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$ROOT.$FILE.PG) LIMIT (3) SCR)
//*
===SEQ FOR PEI
//STEP6  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDUN..$ROOT.$FILE.PP,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=FB,LRECL=110,BLKSIZE=6160)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$ROOT.$FILE.PP) LIMIT (3) SCR)
//*
===SEQ
//
```

4.4.16. DATABASE - TEST DATABASE RESTORATION

3.4. TEST DATABASE RESTORATION

This job executes the REST procedure with, as input, the backup file previously loaded on the disk in STEP4 of job '$prfj.I3I'.

This database is supplied for a few initial tests of opera- tions only. The size of the AN and AR files on the disk is approximately 1,700,000 bytes.

```
===MOD I08REST      Job '$prfj.I4'

STEP1: $radp.REST: test database restoration
```

```
//$PRFJ.I4  JOB ($CCPT),'PAC I04REST',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I04REST                              *
//*    CREATION OF THE VISUALAGE PACBASE TEST DATABASE                *
//*********************************************************************
//TESTREST  EXEC  $RADP.REST
 *TEST
 Y       E
//
```

4.4.17. DATABASE - GENERATION-PRINT REQUEST FILE INIT.

3.5. INITIALIZATION OF THE PRINT-GENERATION REQUEST FILE

The initialization consists of executing the REAG procedure with a request to initialize the AG file (AGI), and, as in- put, an empty AG backup file (PG).

```
===MOD I05REAG      Job '$prfj.I5'

STEP1: $radp.REAG: request initializations
```

```
//$PRFJ.I5  JOB ($CCPT),'PAC I05REAG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              INSTALLATION - I05REAG                               *
//*    INITIALIZATION AND RESTORATION OF THE AG FILE                  *
//*********************************************************************
//*
//STEP1  EXEC $RADP.REAG
 *TEST
 AGI
//PTU560.PAC7PG DD DUMMY,DCB=BLKSIZE=150
//
```

4.4.18. DATABASE - PEI FUNCTION INITIALIZATION

3.6. INITIALIZATION OF THE PRODUCTION ENVIRONEMENT INTERFACE FILES

Initialization of the Production Environment management files, for the PEI variant only. Execution of the INPE and RSPE procedures. (See the Batch Procedures: Administrator's Guide, Chapter VERSIONNING, Subchapter PEI: Production Environment.)

```
===MOD I06PEI      Job '$prfj.I6'

STEP1: $radp.INPE: PEI backup initialization

STEP2: $radp.RSPE: PEI restoration
```

```
===SEQ FOR PEI
//$PRFJ.I6  JOB ($CCPT),'PAC I06PEI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I06PEI                               *
//*  PEI OPTION - PRODUCTION ENVIRONMENT FILE INITIALIZATION          *
//*********************************************************************
//STEP1  EXEC $RADP.INPE
 *TEST
//STEP2  EXEC $RADP.RSPE
 *TEST
//PACR61.PAC7PP DD DSN=$INDUN..$ROOT.$FILE.PP(+1),DISP=SHR
//
===SEQ
```

4.4.19. DATABASE COMPLEMENT - PAC/CS UE INSTALLATION

4. VA PAC DATABASE OPTIONAL INSTALLATION COMPLEMENTS

4.1. DATABASE COMPLEMENT: INSTALLATION OF THE VISUALAGE SMALLTALK DICTIONARY

Loading of VisualAge Smalltalk Objects as User Entities in the VisualAge Pacbase Dictionary.

Execution of the VINS procedure.

```
===MOD I11VGE     Job '$prfj.I11'

STEP1 : $radp.VINS : database updating
```

```
//$PRFJ.I11  JOB ($CCPT),'PAC I11VGE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I11VGE                               *
//*********************************************************************
//VISUAL   EXEC  $RADP.VINS,
//         FDIC='$INDSN..$ROOT.$ROOT.SYEN(VGEDIC)'
 *TEST
//
```

4.4.20. DATABASE COMPLEMENT - PAF FUNCTION INSTALLATION

4.2.1. INSTALLATION OF THE PAF FUNCTION DICTIONARY

The use of the PAF function involves the transformation of SQL requests for access to the VisualAge Pacbase database written in user programs, through the generation of data and VA Pac access sub-program calls in the COBOL source generated from these programs.

The PAF Preprocessor processes the generated programs in order to perform this transformation. It includes the PAFP10 program installed in the batch load-module library MBR8.

Several methods are available to process the generated programs, using PAF:

- with the GPRP procedure, the VA Pac Generation-Printing procedure is followed by the execution of PAF Preprocessor which processes the whole generated flow before sending it to the MVS Internal Reader;

- with the PPAF procedure, the user may:

. 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 methods best suited with the characteristics of the site.

(See the PPAF section in Chapter STANDARD PROCEDURES, subchapter 'GPRT: Generation-Print', in the BATCH PROCEDURES User's Guide.)

Five PAF sub-programs are provided in the installation deck:

. Three batch sub-programs installed in the batch load-module library:

- PBBTST for PAF standard requests
- PBBTWS for PAF requests by keywords
- PBBT98 for physical access to the VA Pac database

NOTE:

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

. Two on-line sub-programs installed in the on-line module library:

- xxTPST for standard PAF requests
- xxTPWS for PAF requests by keyword

The work files necessary for PAF operation are described in the Batch Procedures: User's Guide, Chapter STANDARD PROCEDURES, Subchapter 'GPRT: Generation and Printing', Section 'PPAF...'. Their definition is in the SY parameter PDS: modification of the installation and/or size of these files should be done in this PDS.

An example of the JCL of a user program with PAF calls is supplied in the SYDI Dictionary complements' PDS, in the PAFJCL member. (See PAFJCL, in section 'System: Loading of Parameter PDS', and the description of the example of a user's batch PAF JCL thereafter.)

The work file necessary for the operation of PAF in on-line mode has a CICS-imposed DDNAME of the $root.$root.PA format. This DDNAME must be unique for all programs accessing databases installed under the same root ($ROOT).

Data Element, Data Structure and Segment entities used to write programs involving PAF, are provided as batch transactions in the PAFDIC member of the SYDI Dictionary Complements' PDS.

IMPORTANT:

Loading the 'PAF 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 PAF function, it is recommended to create an independent library network that will be accessed by the site's PAF utilities.

USERS PROGRAMMING WITH COBOL/VS

COBOL/VS PAF sub-programs are gathered in a special library. See Subchapter 'System: Alloc./Loading of Load-Module Library'.

EXAMPLE OF A PAF BATCH USER JCL

The JCL example supplied may be adapted to the site's requirements and facilities. It illustrates the execution of a PAF user batch program but it is not compulsory.

DESCRIPTION OF STEPS

This procedure example contains the following steps:

EXECUTION OF USER PROGRAM

```
.Permanent input files (COMPULSORY):

 -Data file
  PAC7AR:  DSN=&INDUV..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDUV..&ROOT.&FILE.AN
  For better performance, the JCL may be adapted in order
  to implement the batch LSR option.

 -Error message file
  PAC7AE:  DSN=&INDSV..&ROOT.&ROOT.AE

.Input/output work file (COMPULSORY):
 -Query-processing file
  SYSPAF:  DSN=&&SYSPAF

.Printing of error messages
  SYSOUT (COMPULSORY)

.Addition of the user program's specific files
```

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*             - JCL EXAMPLE  -
//*    EXECUTION OF A USER P.A.F. BATCH PROGRAM
//*********************************************************************
//PAFBATCH PROC FILE=$FILE,         NUMBER OF PHYSICAL DATABASE
//      ROOT=$ROOT,        ROOT OF THE PACBASE SYSTEM
//      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='$SCAT',    SYSTEM VSAM CATALOG
//      STEPLIB=,          USER LIBRARY OF LOAD-MODULES
//      OUT=$OUT             OUTPUT CLASS
//*********************************************************************
//MAXKEY  EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&&SYSPAF,DISP=(NEW,KEEP),
//         SPACE=(CYL,(3,3)),
//         LRECL=468,RECORG=KS,KEYOFF=0,KEYLEN=12
//MAXKEY   DD DSN=&INDSN..&ROOT.&ROOT.SY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOT.&ROOT.SY(REPRO999),DISP=SHR
//WITHPAF  EXEC PGM=------
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE   DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//SYSPAF   DD DSN=&&SYSPAF,DISP=(OLD,KEEP)
//------   DD DSN=---
//------   DD DSN=---
//------   DD DSN=---
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//         PEND
//PAFBATCH EXEC PAFBATCH
```

4.4.21. DATABASE COMPLEMENT - PAF+ EXTENSION INSTALLATION

4.2.2. INSTALLATION OF THE PAF+ EXTENSION

The following elements must be installed in order to operate the PAF+ Extension:

. User Entity .PPTEX

. SP and SF skeleton files

. GS user file containing the Extraction Master Paths

User definition of an Extraction Master Path is done through an occurrence of the User entity (UE).

The UE is copied by the installation process in the SYDI PDS member of Dictionary Complements (a member of PAFPTEX). This member contains BATCH transactions which will be input in the Database via the UPDT batch update procedure.

Once you have chosen a library to host the UE, modify the PAFPTEX member with a text editor by filling in the '*' line located at the beginning of the transactions, then execute the UPDT procedure.

The PAF skeletons are installed as a default. They are:

. SP, which translates the User Entity Occurrence into PAF queries.

. SF, which generates a COBOL program. This program, once translated by the PAFP10 program, will be a User Extractor or a Macro-Command called in the Master Outline.

The GS file contains the user Extraction Master Paths. This file is defined in the SY parameters' PDS with the DDname $INDUV.$ROOT.$ROOT.GS.

An example of an execution JCL for a User Extractor can be found in the PTEXJCL member of the SYDI parameter PDS.
See next subchapter for information regarding the execution of a PAF User Extractor.

SUBMISSION OF A PAF EXTENSION USER EXTRACTOR JCL

The JCL example supplied may be adapted to the site's requirements and facilities.

USER INPUT

User input is described in the PAF Reference Manual in Chapter "Execution of a User Extractor / E-type PTEx".

EXECUTION OF THE USER EXTRACTOR: CIBLE

```
.Permanent input files (COMPULSORY):
 -Data file
  PAC7AR:  DSN=&INDUV..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDUV..&ROOT.&FILE.AN
 -Error message file
  PAC7AE:  DSN=&INDSV..&ROOT.&ROOT.AE

.Input/output work file (COMPULSORY):
 -Query processing file
  SYSPAF:  DSN=&&SYSPAF

.Print of system error messages
  SYSOUT  (COMPULSORY)

.Output files:
 -Extraction result (not formatted)
  PAC7SO
 -Extraction result
  PAC7SQ

.Output report:
 -Summary
  PAC7DB
```

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               - JCL EXAMPLE -                                     *
//*    USER EXTRACTOR SUBMISSION JCL                                  *
//*    IN INPUT, ADD A '*' LINE    (USER, PASSWORD, LIBRARY)          *
//*    ********  ADD A X-TYPE LINE (REFER TO THE PAF USER REF. MANUAL)*
//*               COL. 2    :   X                                    *
//*               COL. 3    :   PAF CURSOR                           *
//*               COL. 7    :   UEO CODE                             *
//*               COL. 15   :   START LIMIT                          *
//*               COL. 23   :   END LIMIT                            *
//*               COL. 31   :   DEBUG TYPE (0 OU 1)                  *
//*               COL. 32   :   NUMBER OF PAF BUFFERS                *
//*    CHANGE DCB OF PAC7SQ IN RELATION WITH THE USER ENTITY          *
//*********************************************************************
//PTEXJCL PROC FILE=$FILE,            NUMBER OF PHYSICAL DATABASE
//      ROOT=$ROOT,                     ROOT OF THE VA PAC SYSTEM
//      INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES
//      INDSN='$INDSN',             INDEX OF NON-VSAM SYSTEM FILES
//      INDUV='$INDUV',                   INDEX OF VSAM USER FILES
//*:    VSAMCAT='$VCAT',                        USER VSAM CATALOG
//*:    SYSTCAT='$SCAT',                      SYSTEM VSAM CATALOG
//      STEPLIB=,                   USER LIBRARY OF LOAD-MODULES
//      SORTLIB=,                                 SORT LIBRARY
//      OUT=$OUT,                                 OUTPUT CLASS
//      UWK=$UWK,                                    WORK UNIT
//      SPAMB='(TRK,(1,1),RLSE)',             REQUESTS FILE SPACE
//      SPASQ='(TRK,(5,1),RLSE)',               RESULT FILE SPACE
//      SPAWK='(TRK,(5,1),RLSE)',                 WORK FILE SPACE
//      CYL='(3,1)'                                SORTWORK SPACE
//*********************************************************************
//INPUT  EXEC PGM=PTU001
//*********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&XPAFMB,DISP=(,PASS),
//          UNIT=&UWK,
//          SPACE=&SPAMB,DCB=BLKSIZE=3440
//MAXKEY  EXEC PGM=IDCAMS
//***********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&&SYSPAF,DISP=(NEW,KEEP),
//         SPACE=(CYL,(3,3)),
//         LRECL=468,RECORG=KS,KEYOFF=0,KEYLEN=12
//MAXKEY   DD DSN=&INDSN..&ROOT.&ROOT.SY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOT.&ROOT.SY(REPRO999),DISP=SHR
//*
//-------  EXEC PGM=------
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE   DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//SYSPAF   DD DSN=&&SYSPAF,DISP=(OLD,KEEP)
//PAC7MB   DD DSN=&&XPAFMB,DISP=(OLD,DELETE)
//PAC7SO   DD SPACE=&SPAWK,DCB=BLKSIZE=13080,
//         UNIT=&UWK
//PAC7SQ   DD DSN=&&PAC7SQ,DISP=(,PASS),
//         UNIT=&UWK,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         SPACE=&SPASQ
//PAC7DB   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)
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//         PEND
//PTEXJCL EXEC PTEXJCL
//
```

4.4.22. DATABASE COMPLEMENT - WORKSTATION INSTALLATION

4.3. DATABASE COMPLEMENT: WORKSTATION MAINFRAME COMPLEMENT INSTALLATION

Operating the WorkStation requires the presence on the mainframe of :
- User Entities (UE) and their occurrences supporting the WorkStation entities.
- Methodology Choices.

The UE and their occurrences are stored at installation in the PDS of the SYMT methods, in the following members:

- DESMER for the Merise methodology;
- DESYSM for the YSM methodology.
- DESADM for the SSADM methodology.
- DESOMT for the OMT methodology.
- DESIFW for the IFW methodology.

  For this methodology, the Dictionary must first be pre-loaded with the transactions contained in the DESIFWP file (about 13,500 transactions).

These are batch transactions which will be introduced in the database via the UPDT procedure.

Once the library provided for these entities has been chosen, the '*' line must be entered at the beginning of the member used as input to the UPDT procedure.

The Methodology Choices are also stored at installation in the SYMT parameter PDS, in the following members:

- PARMMER for the Merise methodology;
- PARMYSM for the YSM methodology.
- PARMADM for the SSADM methodology.
- PARMOMT for the OMT methodology.
- PARMIFW for the IFW methodology.

These are batch transactions which will be introduced in the database via the PARM procedure.

The '*' line must be entered at the beginning of the member used as input to the PARM procedure.

IMPORTANT: It is recommended to install only one methodology per sub-network, even though it is technically possible to install several.

4.4.23. DATABASE COMPLEMENT - PQC UE DICTIONARY INSTALLAT.

4.4. INSTALLATION OF THE PQC FACILITY'S USER ENTITY

The PACBENCH QUALITY CONTROL facility (PQC) features a customization option which allows for user-defined quality rules. These rules are specified in the VA Pac Database as occurrences of a dedicated User Entity.

This User Entity is initially supplied in the SYDI system parameter PDS member named PQCUPDT.

This member's contents are formatted in batch update transactions to be used in input by the Database Updating procedure (UPDT).

Once the Library to update is chosen, an '*'-type line must be added in the PQCUPDT member on top of all update lines.

4.4.24. CICS COMPLEMENT - SUBMISSION JCL

5. CICS MODIFICATIONS

5.1. CICS COMPLEMENT: SPECIFICATION OF LOAD-MODULE LIBRARIES IN THE CICS SUBMISSION JCL

Once the CICS tables are updated and compiled and the files correctly defined, the CICS submission JCL must be completed so that all VA Pac system elements are taken into account.

CAUTION:

Be sure to add the DD card of the VA Pac on-line program library in DFHRPL.

If you add the DD card of the batch program library in the same DFHRPL, you will be able to access the VA Pac batch program characteristics on-line (transaction: rrEF, choice: PD). If not, these generation characteristics will be accessible in batch mode only (INSL procedure).

4.4.25. CICS COMPLEMENT - TRANSACTION OUTPUT MODIF.

5.2. MODIFICATION OF THE VA PAC TRANSACTION OUTPUT

Upon exiting the VA Pac 'rrff' transaction, as well as on exiting the 'rrEF' transaction (management of User Parameters and the PEI Function), the system sends a CICS RETURN through the xxR005 program called by XCTL.

You may replace this program by another program, in order to return control to a general user menu or any other manipulation.

For more details, refer to Chapter ENVIRONMENT, Subchapter 'On-Line Environment (CICS)'.

4.4.26. COMPLEMENT - Pac/Impact

6. OTHER INSTALLATION COMPLEMENTS

6.1. COMPLEMENT: INSTALLATION OF Pac/Impact

GDG ALLOCATION

===MOD I20GDG Job '$prfj.I20'

CAUTION!: If SMS is installed in the site, delete the DD //GDGMOD cards from steps GDGBJ and GDGBB before sub- mitting the Job.

STEP1 : IDCAMS : Allocation of GDG for FH file

STEP2 : IEBGENER : Initialization from scratch of FH file

STEP3 : IDCAMS : Allocation of GDG for FO file

STEP4 : IEBGENER : Initialization from scratch of FO file

STEP5 : IDCAMS : Allocation of GDG for FR file

STEP6 : IEBGENER : Initialization from scratch of FR file

(For a description of these files, see Chapter VISUALAGE PACBASE COMPONENTS, Subchapter 'Evolving Files', section 'Pac/Impact Files'.)

```
//$PRFJ.I20 JOB ($CCPT),'PAC I05GDG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*             - IMPACT ANALYSIS -                                   *
//*            INSTALLATION - I05GDG                                  *
//*    BUILDING OF INDEX DATA-GROUP FOR "FH", "FO" AND "FR" FILES     *
//*                                                                   *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS            *
//*********************************************************************
//$RADP.GDG  PROC FILE=$FILE,         NUMBER OF PHYSICAL DATABASE
//         ROOT=$ROOT,                  ROOT OF THE VA PAC SYSTEM
//         USER=,                PACKAGE CODE FOR IMPACT ANALYSIS
//         INDUN='$INDUN',           INDEX OF NON-VSAM USER FILES
//*:       VSAMCAT='$VCAT',                 USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//         STEPLIB='$MODB',             LIBRARY OF LOAD-MODULES
//         OUT=$OUT,                              OUTPUT CLASS
//         DSCB='$DSCB',                     DSCB MODEL FILE
//         VOLS='SER=$VOLUN',              RESULTS FILE VOLUME
//         UNITS=$UNITUN,                    RESULTS FILE UNIT
//         UWK=$UWK                                WORK UNIT
//*********************************************************************
//INIFO1 EXEC PGM=PRMSYS,PARM='&USER'
//***********************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFO,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFO2  EXEC PGM=IDCAMS
//************************
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER.&ROOT.&FILE.FO,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=260,BLKSIZE=5200)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFO,DISP=(OLD,DELETE)
//INIFO3  EXEC PGM=IEBGENER
//*************************
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=260,BLKSIZE=260)
//SYSUT2   DD DSN=&INDUN..&USER.&ROOT.&FILE.FO(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=260,BLKSIZE=26000)
//*
//INIFR1 EXEC PGM=PRMSYS,PARM='&USER'
//***********************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFR,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFR2  EXEC PGM=IDCAMS
//************************
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER.&ROOT.&FILE.FR,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=72,BLKSIZE=21600)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFR,DISP=(OLD,DELETE)
//INIFR3  EXEC PGM=IEBGENER
```

```
//*************************
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=72,BLKSIZE=72)
//SYSUT2   DD DSN=&INDUN..&USER.&ROOT.&FILE.FR(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=72,BLKSIZE=21600)
//*
//INIFQ1 EXEC PGM=PRMSYS,PARM='&USER'
//*********************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFQ,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFQ2  EXEC PGM=IDCAMS
//************************
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER.&ROOT.&FILE.FQ,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=100,BLKSIZE=21600)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFQ,DISP=(OLD,DELETE)
//INIFQ3  EXEC PGM=IEBGENER
//*************************
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=100,BLKSIZE=100)
//SYSUT2   DD DSN=&INDUN..&USER.&ROOT.&FILE.FQ(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=100,BLKSIZE=21600)
//*
//INIFH1 EXEC PGM=PRMSYS,PARM='&USER'
//*********************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFH,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFH2 EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER.&ROOT.&FILE.FH,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=160,BLKSIZE=24000)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFH,DISP=(OLD,DELETE)
//INIFH3  EXEC PGM=IEBGENER
//*************************
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=160,BLKSIZE=160)
//SYSUT2   DD DSN=&INDUN..&USER.&ROOT.&FILE.FH(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=160,BLKSIZE=24000)
//*
// PEND
//GDG   EXEC $RADP.GDG
//INIFO1.PACRIN DD *
  DEFINE GENERATIONDATAGROUP -
```

```
          (NAME ($INDUN..&USER$ROOT.$FILE.FO) LIMIT (3) SCR)
   /*
   //INIFR1.PACRIN DD *
     DEFINE GENERATIONDATAGROUP -
          (NAME ($INDUN..&USER$ROOT.$FILE.FR) LIMIT (3) SCR)
   /*
   //INIFQ1.PACRIN DD *
     DEFINE GENERATIONDATAGROUP -
          (NAME ($INDUN..&USER$ROOT.$FILE.FQ) LIMIT (3) SCR)
   /*
   //INIFH1.PACRIN DD *
     DEFINE GENERATIONDATAGROUP -
          (NAME ($INDUN..&USER$ROOT.$FILE.FH) LIMIT (3) SCR)
   /*
   //
```

4.4.27. COMPLEMENT - PAC/TRANSFER FACILITY

6.2. COMPLEMENT: INSTALLATION OF THE PAC/TRANSFER FACILITY

1. ALLOCATION OF THE PARAMETER FILE

This loading step contains the JOB '$PRFJ.I30', which includes the following steps:

STEP1 : IDCAMS : This step performs the DELETE/DEFINE of the parameter file (UV).

STEP2 : IDCAMS : Initialization of the parameter file (UV).

This process creates in the parameter file a special record without which the file would be useless.

2. ALLOCATION OF THE JOURNAL FILE

This loading process contains the JOB '$PRFJ.I31'.

This job allocates, during the first installation, the compressed journal file (JT) and the Transfer file (TJ).

It includes the following steps:

```
STEP1   : DELETE of the JT and TJ files.
STEP2   : Allocation of the compressed journal file (JT)
          and Transfer file, performed by a IEFBR14.
```

```
//$PRFJ.I30 JOB ($CCPT),'BU VSAM DEF.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER :                                                 *
//*    ALLOCATING THE VSAM UV FILES                                   *
//*     STEP1 : DELETE DEFINE OF UV FILE                              *
//*     STEP2 : INITIALIZING THE PARAMETERS FILE (UV)                 *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN    DD DSN=$INDUN..$ROOT.$FILE.SY(DF$ROOT.$FILE.UV),
//         DISP=SHR
//*
//STEP2    EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//OU1      DD  DSN=$INDUV..$ROOT.$FILE.UV,DISP=SHR
//IN1      DD  DSN=$INDSN..$ROOT.$ROOT.SY(MAXKEY),DISP=SHR
//SYSIN    DD  DSN=$INDSN..$ROOT.$ROOT.SY(REPRO),DISP=SHR
//*
//
```

```
//$PRFJ.I31  JOB ($CCPT),'BU PREP',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*   PAC/TRANSFER :                                                  *
//*   JOB TO BE EXECUTED DURING THE FIRST INSTALLATION ONLY           *
//*     STEP1 : DELETE COMPRESSED JOURNAL FILE                        *
//*     STEP2 : PREALLOCATING COMPRESSED JOURNAL FILE                 *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDUN..$ROOT.$FILE.SY(DL$ROOT.$FILE.JT),
//         DISP=SHR
//         DD DSN=$INDUN..$ROOT.$FILE.SY(DL$ROOT.$FILE.TJ),
//         DISP=SHR
//STEP2    EXEC PGM=IEFBR14
//ALLOCJT  DD DSN=$INDUN..$ROOT.$FILE.JT,
//         DISP=(,CATLG),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179),
//         SPACE=(TRK,(60,15))
//ALLOCTJ  DD DSN=$INDUN..$ROOT.$FILE.TJ,
//         DISP=(,CATLG),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         DCB=(RECFM=FB,LRECL=172,BLKSIZE=8600),
//         SPACE=(TRK,(60,15))
//*
```

4.4.28. COMPLEMENT - VA PAC / TEAMCONNECTION BRIDGE

6.3. COMPLEMENT - VISUALAGE PACBASE/TEAM CONNECTION BRIDGE

6.3.1. ALLOCATION OF SESSION FILE

This loading step is made of the '$PRFJ.I40' job, which includes the following steps:

STEP1: IDCAMS: This step executes a DELETE/DEFINE of the session file (TS).

STEP2: IDCAMS: Initialization of the session file (TS).

This process creates in the session file a special record without which the file would be impossible to use.

6.3.2. INSTALLATION OF THE TEAMCONNECTION DICTIONARY

This loading step is made of the '$PRFJ.I41' job, which includes a call to the VINS procedure.

```
//$PRFJ.I40 JOB ($CCPT),'VAP TEAMC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*        - VISUALAGE PACBASE - TEAMCONNECTION BRIDGE -              *
//*    ALLOCATING THE VSAM TS FILE                                    *
//*     STEP1 : DELETE DEFINE OF TS FILE                              *
//*     STEP2 : INITIALIZING TS FILE                                  *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN    DD DSN=$INDSN..$ROOT.$ROOT.SY(DF$ROOT.$ROOT.TS),
//         DISP=SHR
//*
//STEP2    EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//OU1      DD  DSN=$INDSV..$ROOT.$ROOT.TS,DISP=SHR
//IN1      DD  DSN=$INDSN..$ROOT.$ROOT.SY(MAXKEY),DISP=SHR
//SYSIN    DD  DSN=$INDSN..$ROOT.$ROOT.SY(REPRO),DISP=SHR
//*
//
```

```
//$PRFJ.I41  JOB ($CCPT),'PAC I41TEAM',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION - I41TEAM                              *
//*********************************************************************
//VISUAL   EXEC  $RADP.VINS,
//         FDIC='$INDSN..$ROOT.$ROOT.SYEN(TEAMDIC)'
 *TEST
//
```

4.5. INSTALLATION TESTS

4.5.1. UTILIZATION TESTS

UTILIZATION TESTS

Tests on the operations are divided into three groups:

. Database Utilization Tests,
. Database Management Tests,
. Extraction Utility Tests.

1. VA PAC DATABASE UTILIZATION TESTS

These tests include the following steps:

. Tests on on-line use under CICS,
. Test on batch updating,
. Test on generation and printing of programs.

VA Pac system on-line tests:

- Open the test database files under CICS.

- Test the VA Pac screen branching.

- Execute some updates.

Batch Update Test:

- Execute the '$prfjUPD' JOB (UPDT procedure).

- The VA Pac Database files must be closed under CICS.

Generation-printing Test:

- Execute the '$prfjGPR' JOB (GPRT procedure).

- The VA Pac Database files must be closed under CICS if the Generation-printing request file (AG) is to be read (+AG transaction as input to the GPRT).

4.5.2. DATABASE MANAGEMENT TESTS

2. VA PAC DATABASE MANAGEMENT TESTS

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

They include the following steps, to be executed in this order:

1. Archiving of journal created during the utilization testing phase: execute job '$prfjARC' yielding the PG(1) file.

2. Direct backup of the database: execute job '$prfjSAV' yielding the PC(1) file.

3. Backup of generation-printing requests: execute job '$prfjSVG' yielding the PG file.

4. Library management: addition/deletion of a library: execute job '$prjfMLI' yielding the PC(2) file.

5. Reorganization of the sequential backup, PC(2), of the VA Pac Database: execute job '$prfjREO' yielding the PC(3) file.

6. Reorganization of sequential backup (PG) of the generation-printing request file, and restoration of the AG file: execute job '$prfjREG'.

7. Database restoration using the PJ(1) archived transaction file and the PC(3) database backup file: execute job $prfjRES.

The VA Pac Database files must be closed under CICS when these tests are run.

Once the VA Pac Database is restored, it is advisable to briefly test on-line operations again, after having re-opened the database files.

4.5.3. EXTRACTION-UTILITY TESTS

3. EXTRACTION UTILITY TESTS

The purpose of these tests is to execute the VA Pac Database extraction procedures.

They are made up of the following steps, to be executed in the following order:

1. Extraction of a library as transactions: execute job '$prfj.EXL',
2. Extraction of entities from a library: execute job '$prfj.EXT',
3. Extraction of selected transactions and/or lists of transactions from the archived journal (PJ): execute job '$prfj.EXJ'.

The VA Pac Database files can be open under CICS for all of these tests.

Each of these jobs can be followed by a UPDT to verify the validity of these extracted transactions.

4.5.4. TEST JCL: UPDT

```
//$PRFJ.UPD JOB ($CCPT),'PAC JCLUPDT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLUPDT                           *
//*    TESTING THE BATCH UPDATE                                       *
//*    .THE PACBASE DATABASE FILES MUST BE CLOSED UNDER CICS          *
//*********************************************************************
//UPDT   EXEC $RADP.UPDT
 *TEST            XCN
 0    TEST1       TEST
 P5010   N  TEST                          10BL
 P5010 1 M  AA00 BB00
 P5010 2 A  1 IWA10R
```

4.5.5. TEST JCL: GPRT

```
//$PRFJ.GPR JOB ($CCPT),'PAC JCLGPRT',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLGPRT                           *
//*    TEST FOR PRINTING-GENERATION                                   *
//*********************************************************************
//GPRT   EXEC $RADP.GPRT,USER=TEST
 *TEST            XCN
 Z  LCE
 Z  LCP
/*
//*
```

4.5.6. TEST JCL: ARCH

```
//$PRFJ.ARC JOB ($CCPT),'PAC JCLARCH',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLARCH                           *
//*    TEST OF THE JOURNAL ARCHIVAL PROCEDURE                         *
//*********************************************************************
//ARCH   EXEC $RADP.ARCH
 *TEST
 S
```

4.5.7. TEST JCL: SAVE

```
//$PRFJ.SAV JOB ($CCPT),'PAC JCLSAVE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLSAVE                           *
//*    TESTING THE DATABASE BACKUP                                    *
//*********************************************************************
//SAVE   EXEC $RADP.SAVE
 *TEST
```

4.5.8. TEST JCL: MLIB

```
//$PRFJ.MLI JOB ($CCPT),'PAC JCLMLIB',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLMLIB                           *
//*    TESTING THE LIBRARIAN                                          *
//*********************************************************************
//MLIB   EXEC $RADP.MLIB
 *TEST
 G                        EXISTING DATABASE - NO SIMULATION
C*AP5XCN                  CREATES AP5 LIBRARY (UNDER XCN)
//*
```

4.5.9. TEST JCL: REOR

```
//$PRFJ.REO JOB ($CCPT),'PAC JCLREOR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLREOR                           *
//*    TESTING OF THE DATABASE REORGANIZATION                         *
//*********************************************************************
//REOR   EXEC $RADP.REOR
 *TEST
 V0010
 BAP5
```

4.5.10. TEST JCL: SVAG

```
//$PRFJ.SVG JOB ($CCPT),'PAC JCLSVAG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLSVAG                           *
//*    TESTING THE AG FILE BACKUP                                     *
//*********************************************************************
//SVAG   EXEC $RADP.SVAG
 *TEST
//PTU550.PAC7GP DD DUMMY,DCB=BLKSIZE=150
```

4.5.11. TEST JCL: REAG

```
//$PRFJ.REG JOB ($CCPT),'PAC JCLREAG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLREAG                           *
//*    TESTING THE REORGANIZATION/RESTORE OF 'AG'                     *
//*********************************************************************
//REAG   EXEC $RADP.REAG
 *TEST
 AG
```

4.5.12. TEST JCL: REST

```
//$PRFJ.RES JOB ($CCPT),'PAC JCLREST',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLREST                           *
//*    TESTING THE RESTORATION WITH THE JOURNAL, AFTER REOR.          *
//*                                                                   *
//* 1000 GAPS REQUESTED.                                              *
//* AFTER THE RESTORATION OF THE PACBASE DATABASE, MAKE A FEW         *
//* QUICK TESTS FOR ON-LINE FUNCTIONNING, AFTER HAVING RE-OPENED      *
//* THE PACBASE DATABASE FILES.                                       *
//*********************************************************************
//REST   EXEC $RADP.REST,
//       ARCHJRNL='$INDUN..$ROOT.$FILE.PJ(0)'
 *TEST
 Y01000     REC
//*
```

4.5.13. TEST JCL: EXLI

```
//$PRFJ.EXL JOB ($CCPT),'PAC JCLEXLI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLEXLI                           *
//*    EXTRACTION OF LIBRARIES                                        *
//*********************************************************************
//EXLI   EXEC $RADP.PACX,USER='TEST'
 *TEST            XCN       EXLI1
//*  //UPDT   EXEC $RADP.UPDT
//*  //INPUT.CARTE DD DSN=&&MV,DISP=(OLD,PASS),DCB=BLKSIZE=6160
```

4.5.14. TEST JCL: EXTR

```
//$PRFJ.EXT JOB ($CCPT),'PAC JCLEXTR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLEXTR                           *
//*    EXTRACTION OF ENTITIES                                         *
//*********************************************************************
//EXTR   EXEC $RADP.PACX,USER='TEST'
C*TEST            XCN       EXTR1
 W1EXCEXADFNR
 W1EXCPFL20SL
 W1EXUDTX
 W1EXCSTX10
//*  //UPDT   EXEC $RADP.UPDT
//*   *TEST
//*  //INPUT.CARTE DD DSN=&&MV,DISP=(OLD,PASS),DCB=BLKSIZE=6160
```

4.5.15. TEST JCL: EXPJ

```
//$PRFJ.EXJ JOB ($CCPT),'PAC JCLEXPJ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*               INSTALLATION TEST JCLEXPJ                           *
//*   EXTRACTION FROM ARCHIVED TRANSACTIONS 'PJ' FILE                 *
//*********************************************************************
//EXPJ   EXEC $RADP.PACX,USER='TEST'
 *TEST                      EXPJ1
 JS
```

4.5.16. TEST JCL: UXSR

```
//$PRFJ.UXS JOB ($CCPT),'PAC UXSR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*                INSTALLATION TEST JCLUXSR                          *
//*********************************************************************
//*
//UXSR   EXEC $RADP.UXSR
 *TEST                                          1
C*CEN   CEN
/*
//
```

4.6. UTILIZATION TEST: Pac/Impact

4. TESTS OF THE 'Pac/Impact' FUNCTION

The purpose of these tests is to execute the procedures of the Pac/Impact funtion.

In order to perform the tests, it is recommended to create a separate database for the UXSR procedure.
Initialization of the FP and FQ files must be performed.

The tests include the following steps, which should be executed in this order:

. Definition of entry points: Run the '$prfj.ISE' job, specifying the selection lines, as well as the context.

. Definition of complementary entry points: Run the '$prfj.ISO' job, specifying the selection lines and context. (This step is optional.)

. Printing of entry points: Run the '$prfj.IPE' job.

. Impact analysis: Runt the '$prfj.IAN' job. For the purpose of the tests, one execution of the procedure may be sufficient.

. Result printing: Run the '$prfj.IPI' job, specifying the context as well as desired printouts.

. Printing of impacted criteria: Run the '$prfj.IPQ' job.

4.6.1. TEST JCL: ISEP

```
//$PRFJ.ISE JOB ($CCPT),'PAC ISEP',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              - IMPACT ANALYSIS -                                  *
//*                INSTALLATION TEST JCLISEP                          *
//*********************************************************************
//*
//ISEP   EXEC $RADP.ISEP
 *TEST            ***      E
 EDAT***
/*
//
```

4.6.2. TEST JCL: IPEP

```
//$PRFJ.IPE JOB ($CCPT),'PAC IPEP',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              - IMPACT ANALYSIS -                                  *
//*                INSTALLATION TEST JCLIPEP                          *
//*********************************************************************
//*
//IPEP   EXEC $RADP.IPEP
//
```

4.6.3. TEST JCL: IANA

```
//$PRFJ.IAN JOB ($CCPT),'PAC IANA',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*             -  IMPACT ANALYSIS  -                                 *
//*                INSTALLATION TEST JCLIANA                          *
//*********************************************************************
//*
//IANA   EXEC $RADP.IANA
//
```

4.6.4. TEST JCL: IPIA

```
//$PRFJ.IPI JOB ($CCPT),'PAC IPIA',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              - IMPACT ANALYSIS -                                  *
//*                INSTALLATION TEST JCLIPIA                          *
//*********************************************************************
//*
//IPIA   EXEC $RADP.IPIA
 *TEST            ***      E
 P BIB111111
/*
//
```

4.6.5. TEST JCL: IPFQ

```
//$PRFJ.IPQ JOB ($CCPT),'PAC IPEP',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              - IMPACT ANALYSIS -                                  *
//*                INSTALLATION TEST JCLIPFQ                          *
//*********************************************************************
//*
//IPFQ   EXEC $RADP.IPFQ
//
```

4.6.6. TEST JCL: ISOS

```
//$PRFJ.ISO JOB ($CCPT),'PAC ISOS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*              - IMPACT ANALYSIS -                                  *
//*                INSTALLATION TEST JCLISOS                          *
//*********************************************************************
//*
//ISOS   EXEC $RADP.ISOS
 *TEST            ***      E
 DDATOR
/*
//
```

4.7. UTILIZATION TESTS: PAC/TRANSFER

5. TESTS OF THE PAC/TRANSFER FUNCTION

The purpose of these tests is to execute the procedures of the PAC/TRANSFER versioning management procedures. (For further details, see the BATCH PROCEDURES Operations Manual: Administrator's Guide, Chapter VERSIONING.)

To perform the tests, the VA Pac database Journal must have been archived.

The tests include the following steps, which should be performed in that order:

. Update of the transaction parameters: Execute the '$prfj.RUP' job, specifying the required transaction parameters in the input transaction stream.

. Compression of the archived Journal (optional test): Execute the '$prfj.RJC' job, specifying the name of the archived Journal file to be compressed.

. Creation of the transaction file: Execute the '$prfj.RPF' job, specifying the user code and password, as well as the transaction SET to be built.

. Preparation of the DSMS environment (this test is optional, and necessary only when the VA Pac database is under DSMS control): Execute the '$prfj.RDU' job, specifying the SET to be processed, then run a batch update of the DSMS database with the first stream of transactions generated by the DUPD procedure.

. Generation of transactions: Execute the '$prfj.RRP' job, specifying the user code and password and selecting all the SETs and all the sessions to be processed.

. Update of the VA Pac database by the UPDT procedure with the generated transferred transactions.

. Reinitialization of the DSMS environment (this test is optional, and necessary only when the VA Pac database is under DSMS control): Batch update of the DSMS database with the second stream of transactions generated by the DUPD procedure.

4.7.1. TEST JCL: TRUP

```
//$PRFJ.RUP JOB ($CCPT),'TRUP',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER:  UPDATING THE PARAMETERS FILE                    *
//*********************************************************************
//TRUP    EXEC $RADP.TRUP
 *TEST
 LLLLLGSXXXX        XXXX
 LLLLLGBXXX
 LLLLLGU--------
 LLLLLGC..--------
/*
//
```

4.7.2. TEST JCL: TRPF

```
//$PRFJ.RPF JOB ($CCPT),'TRPF',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER:  CREATING THE TRANSFER FILE                      *
//*********************************************************************
//TRPF EXEC $RADP.TRPF
//INPUT.CARTE DD *
 *TEST
 LTLLLLL
/*
//
```

4.7.3. TEST JCL: TRRP

```
//$PRFJ.RRP JOB ($CCPT),'TRRP',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER : GENERATING THE TRANSFER TRANSACTIONS            *
//*********************************************************************
//TRRP EXEC $RADP.TRRP
//INPUT.CARTE DD *
 *TEST                                LLLLL
/*
//* TRANSFER TRANSACTIONS FOR PACBASE UPDATE (UPDT) INPUT
//PACX.PAC7MV DD DSN=---.---.---,DISP=SHR
//
```

4.7.4. TEST JCL: TRJC

```
//$PRFJ.RJC JOB ($CCPT),'TRJC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER:  COMPRESSING THE ARCHIVED JOURNAL FILE           *
//*********************************************************************
//TRJC    EXEC $RADP.TRJC
//INPUT.CARTE DD *
 *TEST
00
/*
//* ARCHIVED JOURNAL FILE TO COMPRESS
//PTUG05.PAC7PJ DD DSN=---,---,---,DISP=SHR
//*
//
```

4.7.5. TEST JCL: TRDU

```
//$PRFJ.RDU JOB ($CCPT),'TRDU',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER:  PREPARING THE DSMS ENVIRONMENT                  *
//*********************************************************************
//TRDU    EXEC $RADP.TRDU
 *TEST
 LTLLLLL
//*
//* TRANSACTIONS FOR DSMS UPDATE BEFORE EXECUTION OF THE VA PAC
//* UPDT PROCEDURE
//CREAV.SYSUT2 DD DSN=---,---,---,DISP=SHR
//* TRANSACTIONS FOR DSMS UPDATE AFTER EXECUTION OF THE VA PAC
//* UPDT PROCEDURE
//CREAP.SYSUT2 DD DSN=---,---,---,DISP=SHR
//
```

4.8. UTILIZATION TESTS: VA PAC - VA SMALLTALK

UTILIZATION TESTS: VA PAC - VA SMALLTALK INTERFACE

To be able to test the procedures that make up the Interface between VisualAge Pacbase and VisualAge Smalltalk, you need a VA Pac test database, and a file provided by VA Smalltalk (for the VUP1 procedure), compatible with the VA Pac test database.

The procedures must be executed in the following sequence:

. VUP1 (JOB $PRFJ.VU1)

. VUP2 (JOB $PRFJ.VU2)

. VDWN (JOB $PRFJ.VDW)

. VPUR (JOB $PRFJ.VPU)

4.8.1. TEST JCL: VUP1

```
//$PRFJ.VU1 JOB ($CCPT),'VUP1',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//* VA Smalltalk-VA Pac Interface                                     *
//*      CALCULATION OF VA PAC CODES                                  *
//*********************************************************************
//VUP1 EXEC $RADP.VUP1,
//  PBCOD=---.---.---,
//  FICVIS=---.---.---,
//  VISUTIL=---.---.---
//PVA310.PAC7CA DD DSN=---.---.---,DISP=SHR
```

4.8.2. TEST JCL: VUP2

```
//$PRFJ.VU2 JOB ($CCPT),'VUP2',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//* VA Smalltalk-VA Pac Interface                                     *
//*      UPLOAD (GENERATE TRANSACTIONS FOR UPDT)                      *
//*********************************************************************
//VUP2 EXEC $RADP.VUP2,
//     VISUTIL=---.---.---,
//     PBCOD=---.---.---,
//     MVUPDT=---.---.---
```

4.8.3. TEST JCL: VDWN

```
//$PRFJ.VDW JOB ($CCPT),'VDWN',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//* VA Smalltalk-VA Pac Interface                                     *
//* DOWNLOAD                                                          *
//*********************************************************************
//VDWN EXEC $RADP.VDWN,
//     MVGPRT=---.---.---,
//     MVVISUAL=---.---.---
//INPUT.CARTE DD *
 *TEST            BIB
/*
```

4.8.4. TEST JCL: VPUR

```
//$PRFJ.VPU JOB ($CCPT),'VPUR',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*   VA Smalltalk-VA Pac Interface                                   *
//*    PURGE                                                          *
//*********************************************************************
//VPUR EXEC $RADP.VPUR
//INPUT.CARTE DD *
 *UUUUUUUUPPPPPPPP
 SLXXX
 SSNNNNN
/*
//PVA400.PAC7MX DD DSN=---.---.---,DISP=SHR
```

5. INSTALLATION OF A NEW SUB-RELEASE

5.1. STANDARD REINSTALLATION

STANDARD SYSTEM RE-INSTALLATION

Upon receiving a new cartridge (or tape) of the current release of VisualAge Pacbase, containing improvements from the previous sub-release, you must perform a re-installation of the VA Pac system.

The new sub-release is identified by a number, and is delivered in the following form:

. a complete installation tape,

. the list of corrected bugs,

. a set of instructions may also be included to complete the reinstallation instructions given below.

Generally, only program libraries and system files are affected by the new sub-release.

Three different cases may occur:

1. Installation JCL's have been kept

2. Installation JCL's must be regenerated

3. Non-standard re-installation.

1. INSTALLATION JCLs HAVE BEEN KEPT

The STANDARD RE-INSTALLATION procedure consists of executing the jobs contained in the following JCL modules:

```
1) D04MBR : $prfj.D4B job: loading of batch programs
2) D04MTR : $prfj.D4O job: loading on-line programs
3) D04VS  : $prfj.D4V job: loading Cobol/VS sub-programs
4) D06SKEL: $prfj.D6  job: loading skeleton files
5) D07AE0 : $prfj.D7  job: loading error messages
6) D07PARM: $prfj.D7P job: execution of the PARM procedure.
```

1) D04MBR: reloading of batch load-modules (Change the tape name in the VOL=SER= parameter)

NOTE:

This JOB deletes the load-module library, and executes the allocation and the copy of all programs. There are two procedural methods:

A. Complete job execution:

In this case, it is necessary to backup the programs stored in the libraries and which do not come directly from the installation tape (user programs), before executing the job.

B. Execution of the COPY step (IEBCOPY):

In this case, it is recommended that programs which are to be copied be deleted first, in order to avoid disk space problems.

2) D04MTR: reloading of on-line load-modules (Change the tape name in the VOL=SER= parameter)

NOTE:

This JOB deletes the load-module library, and executes the allocation and the copy of all programs. There are two procedural methods:

A. Complete job execution:

In this case, it is necessary to backup the programs stored in the libraries and which do not come directly from the installation tape (user programs), or programs that have been modified (such as xxR000, xxR005 and xxR300), before executing the job;

B. Execution of the COPY step (IEBCOPY):

In this case, it is recommended to have previously deleted programs which are to be copied, in order to avoid space problems.

3) D04VS: reloading of Cobol/VS sub-programs, only if they are used by the Cobol/VS user application programs. (Change the tape name in the VOL=SER= parameter)

4) D06SKEL: reloading of generation skeleton file. (Change the tape name in the VOL=SER= parameter.)

IDCAMS: DELETE/DEFINE and REPRO of the skeleton files.

5) D07AE0: reloading of error messages and documentation file. (Change the tape name in the VOL=SER= paramter.)

Reloading of the AE0 file: using IEBGENER.

6) D07PARM: reloading of error messages and documentation file, with retrieval of the user parameters.

IMPORTANT:

The PARM procedure may require user input, depending on the following:

1. There are no additional user parameters to be added for the reinstallation (no new users, no new VA Pac access keys, etc.), and the parameter backup (generation file whose DSNAME suffix is PE) is valid: the only user input for PARM will be the NRREST command.

2. There are additional user parameters to be added for the re-installation, or the backup is invalid. In this case, user input for PARM will be the NRCHAR command.

2. INSTALLATION JCL's MUST BE REGENERATED FOR A STANDARD RE-INSTALLATION

For detailed information, refer to Subchapters 'Installation Preparation' and "Installation of the complete JCL'.

JCL regeneration consists in re-executing the MM1JCL utility using the parameters provided at installation, and adding the JCL's necessary for reinstallation.

The following JCL module selection lines must be added to the SYSIN:

```
===SELM D04MBR
===SELM D04MTR
===SELM D04VS
===SELM D06SKEL
===SELM D07AE0
===SELM D07PARM
```

Check the resulting JCL's and proceed with the installation according to the steps described in paragraph 1 above.

3. INSTALLATION JCL's MUST BE REGENERATED IN CASE OF A NON-STANDARD RE-INSTALLATION

For detailed information, refer to Subchapters "INSTALLATION PREPARATION" and "INSTALLATION OF THE COMPLETE JCL"

JCL regeneration consists of re-executing the MM1JCL utility using the parameters chosen for installation, and adding the JCL's necessary for reinstallation. These JCL's are indicated in the note enclosed with the sub-release medium.

Once the JCL are regenerated, you may proceed to a standard installation according to the procedure described in paragraph 1 above, and the specific instructions indicated in the note enclosed with the sub-release tape.

6. RETRIEVAL OF PACBASE 802.02, ..., 2.0

6.1. FOREWORD

RETRIEVAL OF PACBASE 8.02v02, 1.5, 1.6, AND 2.0

FOREWORD

Users of the WorkStation, DSMS and Pactables functions should ensure the operating compatibility of these functions with VisualAge Pacbase, according to the versions/releases in use.

The current VisualAge Pacbase release is compatible with:

- WorkStation new release
- DSMS from release 8.0.2 compatible with VA Pac 8.0.2
- Pactables all releases

NOTE:

Users of Pactables 7.3 or 8.0, need a special program, PTA250, for the GETT Pactables procedure. This program is available on request with VisualAge Pacbase Support.

6.2. RETRIEVAL OF PACBASE 2.0

6.2.1. RETRIEVAL OPERATIONS

VISUALAGE PACBASE 2.0 RETRIEVAL OPERATIONS

OPERATIONS TO BE PERFORMED

The installation of the new VA Pac release does not require any particular retrieval of the database files and associated user files, except for the generation-print command file.

Once the new VA Pac release is installed, you must back up the databases and associated user files via the standard procedures of THE RELEASE TO BE RETRIEVED, and restore these databases and files via the standard procedures of THE NEW VA PAC RELEASE. To benefit from the new choices, you should include the reorganization procedure in the retrieval process.

For users of the WorkStation:

The WorkStation's dedicated User Entities must be uploaded into the Database via the UPDT procedure, after the Database has been restored in the new release.

CASE 1: YOUR 2.5 RELEASE IS IN A NEW ENVIRONMENT

1. Reinstallation of user parameters

. Backup of the user parameters, producing a PE file, old release (PARM procedure R 2.0).

. Execution of the new LOAE procedure using as input backup file, the PE file produced by the preceding backup, and the '*********' user code and NRREST command as input parameters.

. Execution of the PARM procedure with, as input, the MBPARM file containing the new key supplied with the product.

. For users of the VA Pac WorkStation, execution of the new PARM procedure, including in the input the transactions associated to the methodolgy used on the site. (See Chapter INSTALLATION, Subchapter 'INSTALLATION PROCESS', Section 'Loading of Error Messages, Documentation, and User Parameters'.) Use the NRCHAR command.

RESULT: AE and AP files, containing the old release user parameters as operational under the new VA Pac release, and the methodology parameters.

2. Reinstallation of a VA Pac Database

. Backup of the database, producing a PC file (old release).

. Journal file initialization (new ARCH procedure).

. Database restoration from the backup resulting from the backup procedure (new REST procedure).

. Backup of generation-print commands, producing a PG file (old release).

. Retrieval of the generation-print commands (RPPG) producing a PG file (new release).

. Restoration of generation-print commands resulting from the backup of the previous step (new REAG procedure).

RESULT: AJ, AN, AR, and AG files operational under the new VisualAge Pacbase Release.

3. Reinstallation of the Production Environment Interface

. PEI backup, producing a PP file (old release).

. PEI restoration (new RSPE procedure) using in input the backup produced by the previous step.

RESULT: AB and AC files, operational under the new VA Pac Release.

CASE 2: YOUR 2.5 RELEASE IS IN THE SAME ENVIRONMENT AS THE 2.0 RELEASE

In this case, there is only one retrieval process to be performed: on the generation-print command file. This file must be backed up by the SVAG 2.0 procedure.

The following steps must be performed:

. Deparameterization of the JCL, according to the installation parameters of the existing release.

. CICS CSD update:

- All maps can be deleted from tables.
- The xxR980 program must be defined.

. Parameter loading:

- D02CPAR: new members DFSYIANA and DLSYIANA.
- D02DICVG and D02DICMT: for Dictionary update.

. Load-module loading:

- D04MBR - Batch programs.
- D04MTR - On-line programs.
- D04VS - Cobol VS programs (if necessary).

. Batch procedure cataloging:

- Two new procedures: IGRA and RPPG.
- Three procedures have been modified: REOR, IANA, and VUP1. (Refer to Subchapter 'JCL differences'.)

. Generation skeleton loading: D06SKEL

. Error message loading: D07AE0, followed by D07PARM (NRCHAR command).

. Database parameter loading: I02SY

Integration of member DFxxyyVP.

. Retrieval of generation-print commands (RPPG) producing a PG file in the new release format.

. Restoration of generation-print commands from the previous backup (REAG procedure of the new release).

. For users of the WorkStation:

- UPDT - Update of the methodology's entities.
- PARM - Update of the methodology's choices.

. For users of the VA Pac/VA Smalltalk Interface:

- I11VGE - Update of user entities.

. For users of the VA Pac/TeamConnection Interface:

- I41TEAM - Update of user entities.

6.2.2. RPPG: RETRIEVAL OF GENERATION-PRINT REQUESTS FILE

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*             - RETRIEVAL OF THE 'PG' FILE -                        *
//*********************************************************************
//$RADP.RPPG PROC FILE=$FILE,         NUMBER OF PHYSICAL DATABASE
//         ROOT=$ROOT,                  ROOT OF THE VA PAC SYSTEM
//         INDUN='$INDUN',            INDEX OF NON-VSAM USER FILES
//*:       VSAMCAT='$VCAT',                 USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',             VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$MODB',                LIBRARY OF LOAD-MODULES
//         UNITS=$UNITUN,                            UNIT OF BACKUP
//         VOLS='SER=$VOLUN',                      VOLUME OF BACKUP
//         DSCB='$DSCB',                            DSCB MODEL FILE
//         OLDPG=,                               DSNAME OF OLD BACKUP
//         SPAPG='(TRK,(20,5),RLSE)',               'SPACE' OF BACKUP
//         OUT=$OUT                                   OUTPUT CLASS
//*********************************************************************
//PTU908 EXEC PGM=PTU908
//*********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7IN   DD DSN=&OLDPG,DISP=SHR
//PAC7OU   DD DSN=&INDUN..&ROOT.&FILE.PG(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=&SPAPG,
//         DCB=(&DSCB,RECFM=FB,LRECL=150,BLKSIZE=6150)
//SYSUDUMP DD SYSOUT=&OUT
```

6.3. RETRIEVAL OF PACBASE 802.02, ..., 1.6

6.3.1. OPERATIONS TO BE PERFORMED

RETRIEVAL OF PACBASE 802v02, ..., 1.6: OPERATIONS

FOREWORD

Users of the WorkStation, DSMS, and Pactables facilities should make sure of the operating compatibility of these products with VisualAge Pacbase, according to the releases in use.

The present VA Pac release supports:

. WorkStation: new release

. DSMS: any release from 8.0.2 --compatible with Pacbase 8.0.2

. Pactables: all releases

NOTE: Users of Pactables release 7.3 or 8.0 need a special PTA250 program to use with the GETT batch procedure. Contact VisualAge Pacbase Support to order this program.

OPERATIONS TO BE PERFORMED

The installation of the new VA Pac release does not require any particular retrieval of the database files and associated user files.

Once the new VA Pac release is installed, you must back up the databases and associated user files via the standard procedures of the release to be retrieved, and restore these databases and files via the standard procedures of the new VA Pac release. To benefit from the new choices, you should include the reorganization procedure in the retrieval process.

For users of the WorkStation:

. The WorkStation's dedicated User Entities must be uploaded into the Database via the UPDT procedure, after the Database has been restored in the new release.

1. Reinstallation of the User parameters

. Backup of the user parameters, producing a PE file, old release (PARM procedure R 8.02 to 1.6)

. Execution of the new LOAE procedure using as input backup file the PE file produced by the preceding backup, and the '********' user code and NRREST command as input parameters.

. Execution of the PARM procedure with, as input, the MBPARM file containing the new key supplied with the product.

. For users of the VA Pac WorkStation, execution of the new PARM procedure, including in the input the transactions associated to the methodolgy used on the site. (See Chapter INSTALLATION, Subchapter 'INSTALLATION PROCESS', Section 'Loading of Error Messages, Documentation, and User Parameters'.) Use the NRCHAR command.

RESULT: AE and AP files, containing the old release user parameters as operational under the new VA Pac release, and the methodology parameters.

2. Reinstallation of a VA Pac Database

. Backup of the database, producing a PC file (old release);

. Journal file initialization (new ARCH procedure).

. Database restoration from the backup resulting from the backup procedure (new REST procedure).

. Backup of generation-print commands, producing a PG file (old release).

. Retrieval of the generation-print commands (RPPG), producing a PG file (new release).

. Restoration of generation-print commands resulting from the backup of the previous step (new REAG procedure).

. Retrieval of sequential archive file (PJ16 procedure). This procedure is optional. It extracts Journal transactions from older archives, using new programs that handle the century in dates.

RESULT: AJ, AN, AR, and AG files operational under the new VisualAge Pacbase Release.

3. Reinstallation of the Production Environment Interface

. PEI backup, producing a PP file (old release).

. Retrieval of sequential backup (PP16 procedure). The retrieval adds the century digits to all dates handled by the PEI Function.

. PEI restoration (new RSPE procedure) using in input the backup produced by the previous step.

RESULT: AB and AC files, operational under the new VA Pac Release.

6.3.2. PJ16: RETRIEVAL OF ARCHIVE JOURNAL

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*        - RETRIEVAL OF 1.6 ARCHIVE JOURNAL BACKUP FOR 2.0 -        *
//*********************************************************************
//$RADP.PJ16  PROC ROOT=$ROOT,         ROOT OF THE PACBASE SYSTEM
//         FILE=$FILE,                   PHYSICAL DATABASE NUMBER
//         INDUN='$INDUN',         INDEX OF NON-VSAM SYSTEM FILES
//         STEPLIB='$MODB',               LIBRARY OF LOAD-MODULES
//         OUT=$OUT,                                OUTPUT CLASS
//         PJ16=,           DSNAME OF 1.6 ARCHIVAL JOURNAL BACKUP
//         VOLS='SER=$VOLUN',               JOURNAL BACKUP VOLUME
//         UNITS=$UNITUN,         BACKUP UNIT (DISK OR CARTRIDGE)
//         DSCB='$DSCB',                        DSCB MODEL FILE
//         SPAPJ='(TRK,(10,1),RLSE)'             BACKUP 'SPACE'
//*********************************************************************
//REP2PJ EXEC PGM=REP2PJ
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7PJ   DD DSN=&PJ16,DISP=SHR
//PAC7JP   DD DSN=&INDUN..&ROOT.&FILE.PJ(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,
//            VOL=&VOLS,
//            SPACE=&SPAPJ,
//            DCB=(&DSCB,RECFM=FB,LRECL=167,BLKSIZE=6179)
```

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*             - PEI OPTION : TRANSFORMATION OF 1.6 FILES -          *
//*********************************************************************
//$RADP.PP16 PROC ROOT=$ROOT,        ROOT OF THE PACBASE SYSTEM
//         FILE=$FILE,               NUMBER OF PHYSICAL DATABASE,
//         INDUN='$INDUN',           INDEX OF USER NON VSAM FILES
//         OLDPP='NULLFILE',         1.6 PEI FILES BACKUP DSNAME
//*:       VSAMCAT='$VCAT',          USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',          PACBASE SYSTEM VSAM CATALOG
//         STEPLIB='$MODB',   LIBRARY OF LOAD-MODULES
//         OUT=$OUT,                 OUTPUT CLASS
//         VOLS='SER=$VOLUN',    BACKUP VOLUME
//         UNITS=$UNITUN,        BACKUP UNIT (DISK OR CARTRIDGE)
//         DSCB='$DSCB',             DSCB MODEL FILE
//         SPAPP='(TRK,(5,2),RLSE)'  BACKUP SPACE (IF DISK)
//*********************************************************************
//PACR90 EXEC PGM=PACR90
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7PE   DD DSN=&OLDPP,DISP=SHR
//PAC7PS   DD DSN=&INDUN..&ROOT.&FILE.PP(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=&SPAPP,
//         DCB=(&DSCB,LRECL=110,RECFM=FB,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT
```

6.3.3. TRRT (PAC/TRANSFER FACILITY)
6.3.3.1. TRRT: INTRODUCTION

TRRT: RETRIEVAL OF PARAMETERS FILE - OPERATIONS TO PERFORM

PRESENTATION OF TRRT

In releases earlier than 1.6, only one set of parameters could be stored in the UV Parameters file.
To define another Transaction Set, the duplication of the parameters was necessary. According to needs, the procedures' execution JCL had to be adapted to use different Parameters files.

It is now possible to store several Sets of parameters in a single file.

>>>> In any case, the format of UV Parameters files earlier than Rel. 1.6 is not compatible with Pac/Transfer 2.0. This is why the TRRT procedure must be executed on all 'old' UV files.

OPERATING MODE

You may use the TRUP procedure which creates the 1.6 UV Parameters file, defining all Transactions Sets. In this case, you will have to reenter information already entered in your older file(s).

If the number of files to process is high, the operation may imply a substantial workload. This is when the TRRT utility comes in handy.
For each former UV file, TRRT generates parameters in the adequate format, under a Transaction Set code you have specified in input.

NOTE: One TRRT execution can process one former UV file only. You must run TRRT as many times as there are 'old' UV files.

Once all former UV files are processed, use these generated parameters in input to the TRUP procedure.

NOTE: If you run a single TRUP execution including all Transaction Sets, make sure that each Set comes in with a distinct code.

As a result, you have an up-to-date UV Parameters file including all your Transaction Sets.

6.3.3.2. TRRT: USER INPUT

USER INPUT

. User identification line (required)

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

. Definition of Transaction Set (required)

Pos.	Len.	Value	Significance
2	2	'LT'	Line code
3	5	llll	Transaction Set code

6.3.3.3. TRRT: DESCRIPTION OF STEPS

TRRT: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

SEQUENTIAL COPY OF OLDER UV FILE: IDCAMS

.Input file:
 -Former-release UV file
  IN1 : DSN= UV802 parameter of the procedure

.Output file:
 -Former-release UV file, sequential format
  OUT1: DSN=&&TRRTUA

CREATION OF TRANSACTIONS FOR TRUP: PTUG90

This step generates transactions associated to the creation
of the UV file, rel. 2.0.

.Permanent input files:
 -Data file
  PAC7AR: DSN=&INDUVX..&ROOT.&FILE.AR
 -Error messages
  PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
 -2.0 parameter file
  PAC7UV: DSN=&INDUV..&ROOT.&FILE.UV
 -Older, sequential, UV file
  PAC7UA: DSN=&&TRRTUA

.Transaction file:
 -User input
  PAC7MB: DSN=&&TRRTMB

.Output file:
 -Transactions associated to the update of the UV 2.0
  file for TRUP
  PAC7MU: DSN=&&TRRTMU

.Output reports:
 -List of entries
  PAC7ET
 -User check
  PAC7DD

6.3.3.4. TRRT: EXECUTION JCL

```
//*********************************************************************
//* VisualAge Pacbase                                                 *
//*                                                                   *
//*    PAC/TRANSFER:                                                  *
//*    RETRIEVAL OF THE UV PARAMETER FILE, REL. OLDER THAN 1.6        *
//*********************************************************************
//$RADP.TRRT PROC FILE=$FILE,         NUMBER OF PHYSICAL DATABASE
//    ROOT=$ROOT,                      ROOT OF THE PACBASE SYSTEM
//    INDSVX='$INDSVX',         INDEX OF PACBASE SYSTEM FILE (AE)
//    INDUV='$INDUV',                      INDEX OF VSAM USER FILE
//    INDSV='$INDSV',                    INDEX OF VSAM SYSTEM FILE
//    INDUN='$INDUN',                       INDEX OF NON VSAM FILE
//    OUT=$OUT,                                    OUTPUT CLASS
//*:  VSAMCAT='$VCAT',                         USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',                PACBASE SYSTEM VSAM CATALOG
//    STEPLIB='$MODB',                    LIBRARY OF LOAD-MODULES
//    UV802=,               DSN OF UV FILE TO BE RETRIEVED (8.02)
//    SPAUA='(TRK,(10,5),RLSE)',        TEMPORARY 'UA' FILE SPACE
//    SPAMU='(TRK,(10,5),RLSE)',        TEMPORARY 'MU' FILE SPACE
//    UWK=$UWK                                          WORK UNIT
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//***********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TRRTMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//REUV802  EXEC PGM=IDCAMS
//************************
//IN1      DD DSN=&UV802,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDUN..&ROOT.&FILE.SY(REPRO),DISP=SHR
//OU1      DD DSN=&&TRRTUA,DISP=(,PASS),UNIT=&UWK,
//         SPACE=&SPAUA,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
//PTUG90   EXEC PGM=PTUG90
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AR   DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE   DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7UV   DD DSN=&INDUV..&ROOT.&FILE.UV,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&TRRTMB,DISP=(OLD,PASS)
//PAC7UA   DD DSN=&&TRRTUA,DISP=(OLD,PASS)
//PAC7MU   DD DSN=&&TRRTMU,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMU,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
```

6.4. JCL DIFFERENCES

LIST OF PROCEDURES MODIFIED FOR RELEASE 2.5

```
REOR :

       Addition of step PTU2CL

I - //PTU200 EXEC PGM=PTU200,COND=(0,NE,PTU2CL)
D - //PTU200 EXEC PGM=PTU200
I - //PAC7MB   DD DSN=&&REORBM,DISP=(OLD,PASS)
D - //PAC7MB   DD DSN=&&REORMB,DISP=(OLD,PASS)

I - //SORTQS EXEC PGM=SORT,COND=((0,NE,PTU2CL),(0,NE,PTU200)
D - //SORTQS EXEC PGM=SORT,COND=(0,NE,PTU200)

I - //PTU208 EXEC PGM=PTU208,
I - //    COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS))
D - //PTU208 EXEC PGM=PTU208,COND=((0,NE,PTU200),(0,NE,SORTQ
I - //PAC7MB   DD DSN=&&REORBM,DISP=(OLD,PASS)
D - //PAC7MB   DD DSN=&&REORMB,DISP=(OLD,PASS)

I - //PTU210 EXEC PGM=PTU210,
I - //    COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS))
D - //PTU210 EXEC PGM=PTU210,COND=((0,NE,PTU200),(0,NE,SORTQ
I - //PAC7MB   DD DSN=&&REORBM,DISP=(OLD,PASS)
D - //PAC7MB   DD DSN=&&REORMB,DISP=(OLD,PASS)

step PTU220
I - //   COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
                (0,NE,PTU210))
D - //   COND=((0,NE,PTU200),(0,NE,SORTQS),(0,NE,PTU210))

step SORTAN
I - //   COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
                (0,NE,PTU210))
D - //   COND=((0,NE,PTU200),(0,NE,SORTQS),(0,NE,PTU210))

I - //PTU240 EXEC PGM=PTU240,COND=((0,NE,PTU2CL),(0,NE,PTU20
I - //      (0,NE,SORTQS),(0,NE,PTU210),(0,NE,SORTAN))
D - //PTU240 EXEC PGM=PTU240,COND=((0,NE,PTU200),
D - //      (0,NE,SORTQS),(0,NE,PTU210),(0,NE,SORTAN))
I - //PAC7MB   DD DSN=&&REORBM,DISP=(OLD,DELETE)
D - //PAC7MB   DD DSN=&&REORMB,DISP=(OLD,DELETE)


IANA :

For sites where SMS is not installed, the FQ file is no
longer a temporary file.
```

VDWN, VPUR, VUP1, and VUP2:

Use of BLSR for optimization.
Addition of a VERIFY on the database files.
The VC file is replaced by the VP file.

LIST OF PROCEDURES MODIFIED FOR RELEASE 2.0

```
ARCH - CAUTION: user entry has been modified

 step PTU300
I - //           DCB=(&DSCB,RECFM=FB,LRECL=167,BLKSIZE=6179)
D - //           DCB=(&DSCB,RECFM=FB,LRECL=165,BLKSIZE=6105)
I - //PAC7PQ   DD DUMMY,DCB=BLKSIZE=167
D - //PAC7PQ   DD DUMMY,DCB=BLKSIZE=165


CSES - Support of Pactables files

 step PTUCSS
I - //          DCB=(&DSCB,RECFM=FB,BLKSIZE=6179,LRECL=167)
D - //          DCB=(&DSCB,RECFM=FB,BLKSIZE=6105,LRECL=165)

I - //PAC7TC   DD DSN=&DSNTC0,DISP=SHR
I - //PAC7CT   DD DSN=&DSNTC1,
I - //          DISP=(,CATLG,DELETE),
I - //          UNIT=&UNITS,VOL=&VOLS,
I - //          SPACE=&SPATC,
I - //          DCB=(&DSCB,RECFM=FB,BLKSIZE=10614,LRECL=1061)


ESES :

  step PTUESS
I - //PAC7AN   DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
I - //PAC7AR   DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR

D - //PAC7PC   DD DSN=&INDUN..&ROOT.&FILE.PC(0),DISP=SHR


GETA : compatibility with 1.2 tables

       Addition of step PACT45


GETD : compatibility with 1.2 tables

       Addition of step PACT45
```

```
GPRx :

I - //PAC    EXEC PGM=PACB,REGION=0K
D - //PAC    EXEC PGM=PACB&LNG,REGION=0K

I - //PAC7G6   DD DSN=&&PAC7G6,DISP=(,PASS),UNIT=&UWK,
I - //            DCB=(RECFM=FB,LRECL=345,BLKSIZE=13800),
I - //            SPACE=&SPAMAN


HIPE :

I - //         DCB=(LRECL=167,RECFM=FB,BLKSIZE=6179)
D - //         DCB=(LRECL=165,RECFM=FB,BLKSIZE=6105)


PQCE - Replacement of step PTUUSE with PACX

I - //MAXKEY EXEC PGM=IDCAMS
I - //PACX   EXEC PGM=PACX,REGION=0K
D - //PTUUSE EXEC PGM=PTUUSE

I - //PTUQ10 EXEC PGM=PTUQ10,COND=(0,NE,PACX)
D - //PTUQ10 EXEC PGM=PTUQ10,COND=(0,NE,PTUUSE)

I - //PTUQ15 EXEC PGM=PTUQ15,COND=((0,NE,PACX),...
D - //PTUQ15 EXEC PGM=PTUQ15,COND=((0,NE,PTUUSE),...


REOR :

I -//PTU208 EXEC PGM=PTU208,COND=((0,NE,PTU200),(0,NE,SORTQS
D -//PTU208 EXEC PGM=PTU208,COND=(0,NE,PTU200)

I -//PTU210 EXEC PGM=PTU210,COND=((0,NE,PTU200),(0,NE,SORTQS
D -//PTU210 EXEC PGM=PTU210,COND=(0,NE,PTU200)

I -//PTU220 EXEC PGM=PTU220,REGION=4096K,
I -//   COND=((0,NE,PTU200),(0,NE,SORTQS),(0,NE,PTU210))
D -//PTU220 EXEC PGM=PTU220,
D -//          COND=((0,NE,PTU200),(0,NE,PTU210)),REGION=20

I -//SORTAN EXEC PGM=SORT,
I -//   COND=((0,NE,PTU200),(0,NE,SORTQS),(0,NE,PTU210))
D -//SORTAN EXEC PGM=SORT,COND=((0,NE,PTU200),(0,NE,PTU210)

I -//PTU240 EXEC PGM=PTU240,
I -//   COND=((0,NE,PTU200),(0,NE,SORTQS),...,(0,SORTAN))
D -//PTU240 EXEC PGM=PTU240,COND=((0,NE,PTU200),...
```

REST and RESY:

```
 step PTU420
I - //PAC7JO   DD DSN=&ARCHJRNL,DISP=OLD,DCB=BLKSIZE=6179
D - //PAC7JO   DD DSN=&ARCHJRNL,DISP=OLD,DCB=BLKSIZE=6105

 step PACA15
I - //            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
D - //            DCB=(RECFM=FB,LRECL=165,BLKSIZE=6105)
```

UPDP and UPDT:

```
 step PACA05 or PAF900
I - //            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
D - //            DCB=(RECFM=FB,LRECL=165,BLKSIZE=6105)

 step PACA15
I - //            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
D - //            DCB=(RECFM=FB,LRECL=165,BLKSIZE=6105)
```

LIST OF NEW PROCEDURES

Proc.	Comments	Rel.
GET0	Compatibility with Pactables 1.2	2.0
GET1	'' ''	''
GET2	'' ''	''
IANA	Module Pac/Impact	2.0
IGRA	'' '' ''	2.5
INFP	'' '' ''	2.0
INFQ	'' '' ''	''
IPEP	'' '' ''	''
IPFQ	'' '' ''	''
IPIA	'' '' ''	''
ISEP	'' '' ''	''
ISOS	'' '' ''	''
PACX	Deep extractors	2.0
PJ16	Retrieval of 1.6 Journal	2.0
PP16	Retrieval of 1.6 PEI files	''
RPPG	Retrieval of 2.0 PG file	2.5
PRGS	Printout of master-path file	2.0
TCCI	VA Pac-TeamConnection Bridge	2.0
TCGP	'' '' ''	''
TCLS	'' '' ''	''
TCME	'' '' ''	''
TRDU	Pac/Transfer	2.0
TRJC	''	''
TRPF	''	''
TRRP	''	''
TRRT	''	''
TRUP	''	''
UXSR	Sub-network extraction	2.0
VDWN	VA Pac-VA Smalltalk Bridge	2.0
VPUR	'' '' ''	''
VUP1	'' '' ''	''
VUP2	'' '' ''	''

LIST OF PROCEDURES DELETED SINCE RELEASE 2.0

Proc.	Program(s)	Comments
EXLI	PTU800	Replaced with procedure PACX
EXTR	PACS10	" "
EXPJ	PTU600 PTU610	" "
EXPU	PTU880 PTU885	" "
RMEN	PTU860 PTU865	" "
"	PTU866	
EXUE	PTUUSE	" "
EXSN	PTU840	" "
UPAE	PTUMAE	Integrated in procedure PARM
CSEP		Integrated in procedure CSES
TRDQ	PTUDQ2	Not maintained
ECSP	PTUCSP	Not maintained
DCOB	PTUCR1 PTUCR2	Not maintained
"	PTUD10 PTUD20	
"	PTUD30	
Retrieval procedures (If necessary, contact the Support.)		
RP6A	PTU930	Retrieval PAC700
RP6B		" "
EX62	PTU830	" "
PC73	PTURPC	Retrieval 7.3
PE73	PTU902	" "
PJ73	PTU916	" "
PP73	PACR02	" "
PC80	REPGDP	Retrieval 8.0
PE80		" "
PJ80	PTU917	" "
PCYS	REPYSM	Retrieval YSM methodology
PJYS	REJYSM	" "
RTYS	REPAFL	" "
TRUV	PTU890	Retrieval U manuals into V manuals