

VisualAge Pacbase



The Developer's Procedures Z/OS CICS

Version 3.5



VisualAge Pacbase



The Developer's Procedures Z/OS CICS

Version 3.5

Note

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

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

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

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

Ninth Edition (November 2011)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.5

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

IBM France Software Laboratory, Rational Division
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.

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

© Copyright IBM Corporation 1983,2011.

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

Contents

| | | | |
|--|------------|---|------------|
| Notices | vii | GPMC - Management of MCI Operator | 61 |
| Trademarks | ix | GPMC - Introduction | 61 |
| Chapter 1. General Introduction to the Batch Procedures | 1 | GPMC - User Input | 62 |
| Foreword | 1 | GPMC - Description of Steps | 62 |
| Overview of the Procedures | 1 | GPMC - Execution JCL | 63 |
| User Identification '*' Line. | 2 | Chapter 3. Extractions | 71 |
| Access Authorizations | 3 | PACX - Introduction | 71 |
| Abnormal Endings | 3 | PACX - User Input Common to all Extractors | 71 |
| Performances | 5 | EXTR/EXTA - Extraction of Entities | 73 |
| Chapter 2. Generation and Printing | 7 | EXTR/EXTA - Introduction | 73 |
| GPRT - The Generation/Print Procedure | 7 | EXTR/EXTA - User Input | 73 |
| GPRT - Introduction. | 7 | EXUE - Extraction of User Entities Contents | 76 |
| GPRT - User Input / Results | 10 | EXUE - Introduction | 76 |
| GPRT - Generation / Print Commands | 10 | EXUE - User Input. | 76 |
| GPRT - Processing of Job Streams | 25 | PACX - Description of Steps | 77 |
| GPRT - Description of Steps. | 25 | PACX - Execution JCL | 79 |
| GPRT - Execution JCL. | 28 | Chapter 4. Personalized Extraction/Automated Documentation | 85 |
| GPRU - Execution JCL | 33 | Foreword | 85 |
| GPRP - Execution JCL. | 39 | Personalized Extractions - PAF+ | 85 |
| EMLD - Loading of User-Defined Error | | XPAF - Validation of an Extraction Master | |
| Messages | 44 | Path. | 85 |
| EMLD - Introduction | 44 | XPAF - Introduction | 85 |
| EMLD - User Input | 44 | XPAF - User Input | 86 |
| EMLD - Description of Steps | 45 | XPAF - Description of Steps. | 87 |
| EMLD - Execution JCL | 45 | XPAF - Execution JCL. | 89 |
| EMUP - Update of User-Defined Error | | XPAF - Operation of an Extraction Master | |
| Messages | 46 | Path. | 92 |
| EMUP - Introduction | 46 | Documentation Structuring - PDM+ | 92 |
| EMUP - User Input | 47 | XPDM - Validation of a Master Outline | 92 |
| EMUP - Description of Steps | 47 | XPDM - Introduction | 92 |
| EMUP - Execution JCL | 48 | XPDM - User Input | 93 |
| PPAF - Generated Programs PAF Preprocessor | 49 | XPDM - Description of Steps | 94 |
| PPAF - Introduction | 49 | XPDM - Execution JCL | 95 |
| PPAF - User Input | 50 | Extraction Master Path and Outline File. | 97 |
| PPAF - Description of Steps. | 51 | PRGS - Printing of Master Path / Outline | |
| PPAF - Execution JCL. | 51 | File | 97 |
| GPRC - COBOL API management. | 52 | PRGS - Introduction | 97 |
| GPRC - Introduction | 52 | PRGS - User Input | 97 |
| GPRC - User Input. | 53 | PRGS - Description of Steps. | 97 |
| GPRC - Description of Steps | 53 | PRGS - Execution JCL. | 98 |
| GPRC - Execution JCL | 54 | Chapter 5. Batch Update | 101 |

| | | | |
|--|-----|---|-----|
| UPDP - Update from PAF Tables. | 101 | On-Line Screens | 136 |
| UPDP - Introduction | 101 | Definition (Line H) | 136 |
| UPDP - User Input / Update Rules / | | Dialog Complement (Line H3) | 138 |
| Results | 101 | Description (Line I) | 139 |
| UPDP - Description of Steps | 103 | Call of Segments (Line H2) | 141 |
| UPDP - Execution JCL | 105 | Call of Macro-Structures (Line M) | 142 |
| UPDT - Update | 108 | Program Beginning Insertions (Line | |
| UPDT - Introduction | 108 | D) | 143 |
| UPDT - User Input / Update Rules / | | Working Areas (Line 7) | 143 |
| Results | 109 | Procedural Code (Line P) | 146 |
| Multi-entity User Input | 111 | Programs | 146 |
| Multi-purpose Line (Line VC, VG, | | Definition (Line 0) | 146 |
| VE, VO) | 111 | Call of Data Structures (Line 1) | 147 |
| Parameterized Input Aids/Variable | | Call of Macro-Structures (Line M) | 148 |
| Parts (Line VZ) | 112 | Program Beginning Insertions (Line | |
| Call of Instances via Relations (Line | | D) | 149 |
| QR) | 113 | Working Areas (Line 7) | 149 |
| Entity Update Lock (Line R) | 114 | Procedural Code (Line P) | 152 |
| Search by Keywords (Line G) | 115 | COBOL Source Lines (Line FC) | 152 |
| Data Elements | 116 | Pure COBOL Source Lines (Line 9) | 153 |
| Definition (Line C) | 116 | Database Blocks (Hierarchical) | 153 |
| Description (Line E) | 117 | Definition (Line L1) | 153 |
| Model Objects | 118 | Description (Line L2) | 154 |
| Definition (Line K1) | 118 | Database Blocks (Codasyl) | 155 |
| Call of Properties in Object or Relat. | | Definition (Line L1) | 155 |
| (Line K3). | 118 | Description (Line L3) | 155 |
| Model Relations | 119 | Database Blocks (Relational-SQL) | 156 |
| Definition (Line K1) | 119 | Definition (Line L1) | 156 |
| Call of Objects in Relation or F.I.C | | Description (Line L4) | 156 |
| (Line K2). | 120 | Database Blocks (Turboimage) | 158 |
| Call of Properties in Object or Relat. | | Definition (Line L1) | 158 |
| (Line K3). | 120 | Description (Line L2) | 158 |
| Model F.I.C.'s | 121 | Texts | 159 |
| Definition (Line K1) | 121 | Definition (Line S) | 159 |
| Call of Objects in Relation or F.I.C | | Description (Line T) | 159 |
| (Line K2). | 121 | Documents | 161 |
| Data Structures | 122 | Definition (Line W1) | 161 |
| Definition (Line A) | 122 | Description (Line W2) | 162 |
| Segments | 122 | Parameterized Input Aids | 163 |
| Definition (Line 2) | 122 | Definition (Line V1) | 163 |
| Description (Line 3) | 126 | Description (Line V2) | 163 |
| Pactables Sub-Schemas and | | Meta-Entities | 164 |
| Sub-Systems (Line 21) | 127 | Definition (Line Y1) | 164 |
| Reports | 127 | Detail Line Definition (Line Y6) | 164 |
| Definition (Line B) | 127 | Description (Line Y2) | 165 |
| Report Layout Description (Line 4) | 128 | User-Defined Relations | 165 |
| Report Characteristics Description | | Definition (Line Y5) | 165 |
| (Lines 5, E) | 128 | Client User Entities | 166 |
| List of Categories (Line 5) | 130 | Definition (Line Y3) | 166 |
| Description of Structures (Line 6) | 134 | Description (Line Y4) | 166 |

| | | | |
|---|------------|---|------------|
| Extension User Entities | 167 | ISOS - Description of Steps | 203 |
| Definition (Line YC) | 167 | ISOS - Execution JCL | 204 |
| Description (Line YD) | 167 | IMFH - Merge of FH Files - Creation of FH | |
| Thesaurus | 168 | and FR | 207 |
| Enrichment of the Thesaurus (Line | | IMFH - Introduction | 207 |
| G1) | 168 | IMFH - Description of Steps | 207 |
| Library | 168 | IMFH - Execution JCL | 208 |
| Definition (Line X) | 168 | INFQ - FQ File Reinitialization (Impact | |
| UPDT - Description of Steps | 169 | Analysis). | 209 |
| UPDT - Execution JCL | 171 | INFQ - Introduction | 209 |
| Chapter 6. Pactables | 175 | INFQ - Description of Steps | 209 |
| GETD-GETA - Description Generators | 175 | INFQ - Execution JCL | 210 |
| GETD-GETA - Introduction | 175 | IGRA - Breaking down of Group Fields | 211 |
| GETD-GETA - User Input / Result | 176 | IGRA - Introduction | 211 |
| GETD-GETA - Description of Steps | 177 | IGRA - Description of Steps | 213 |
| GETD - Execution JCL | 178 | IGRA - Execution JCL | 215 |
| GETA - Execution JCL | 181 | IANA - Impact Search Criteria | 218 |
| GETI - Initialization of Description Line | 183 | IANA - Introduction | 218 |
| GETI - Introduction | 183 | IANA - Description of Steps | 219 |
| GETI - User Input. | 183 | IANA - Execution JCL | 223 |
| GETI - Description of Steps | 183 | IPFQ - FQ File Printout (Impact Analysis) | 227 |
| GETI - Execution JCL | 184 | IPFQ - Introduction | 227 |
| SMTD-RMTD - Migration of Tables | | IPFQ - User Input. | 228 |
| Descriptions | 185 | IPFQ - Description of Steps | 229 |
| SMTD - Introduction. | 185 | IPFQ - Execution JCL | 230 |
| SMTD - Description of Steps | 186 | IPEP - Entry Points Printout | 232 |
| SMTD - Execution JCL | 186 | IPEP - Introduction | 232 |
| RMTD - Introduction | 187 | IPEP - Description of Steps. | 233 |
| RMTD - Description of Steps | 187 | IPEP - Execution JCL. | 233 |
| RMTD - Execution JCL | 188 | IPIA - Printing of the Impact Analysis | |
| Chapter 7. Pac/Impact. | 189 | Results | 234 |
| Foreword | 189 | IPIA - Introduction | 234 |
| INFP - FP File Initialization (Impact | | IPIA - User Input | 235 |
| Analysis). | 189 | IPIA - Description of Steps. | 237 |
| INFP - Introduction | 189 | IPIA - Execution JCL. | 239 |
| INFP - User Input. | 190 | Chapter 8. Methodology Integrity Check | 241 |
| INFP - Description of Steps | 190 | ADM - SSADM Pacdesign Methodology | 241 |
| INFP - Execution JCL | 191 | SADM - Introduction | 241 |
| ISEP - Selection of Entry Points | 193 | SADM - User Input | 241 |
| ISEP - Introduction | 193 | SADM - Description of Steps | 242 |
| ISEP - User Input | 194 | SADM - Execution JCL | 243 |
| ISEP - Description of Steps. | 196 | YSMC - YSM Methodology / WorkStation | 246 |
| ISEP - Execution JCL. | 197 | YSMC - Introduction. | 246 |
| ISOS - Selection of Strings and Operators | 199 | YSMC - User Input | 246 |
| ISOS - Introduction | 199 | YSMC - Description of Steps | 248 |
| ISOS - User Input. | 201 | YSMC - Execution JCL | 250 |

Notices

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

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

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

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

Trademarks

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

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

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

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

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

Chapter 1. General Introduction to the Batch Procedures

Foreword

This manual documents the batch procedures that all the product users are likely to use.

These procedures first include all standard procedures dedicated to updating, generating, printing, and extracting.

They also include the procedures dedicated to the following functionalities:

- Personalized extraction and automated documentation,
- Integrity checks on Methodology occurrences (associated with the VA Pac WorkStation's Pacdesign module for SSADM and YSM),
- Pac/Impact.

Overview of the Procedures

Batch processes are grouped into procedures. The objective of the following chapters is to present each of the procedures that are likely to be used, and to specify their execution conditions.

The following elements are included for each procedure:

- a general introduction including:
 - the Execution Conditions,
 - operations to be performed in case of Abnormal Executions.
- the description of the User Input, Processes and Results obtained, possibly including use recommendations.
- the Description of Steps.

To use a procedure on a given Database, the user must have the corresponding authorization.

Each user has:

- a general level of authorizations to the batch procedures,
- a specific authorization level per Database.

User authorizations are defined in the Administration Database.

User Identification '*' Line

Batch procedures which access the Databases require a user identification ('*-type) line at the beginning of user input to identify the user as well as the Library and session in which he/she wishes to work.

Some information entered on this line is the same as that entered on the Sign-on screen. It is thus possible to check if the user's commands are compatible with his/her authorizations.

Before running any batch procedure, the user must make sure he/she has the adequate authorization level.

| Position | Length | Value | Meaning |
|----------|--------|----------|---|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | Password |
| 19 | 3 | bbb | Library code |
| 22 | 4 | nnnn | Session number (space if current session) |
| 26 | 1 | | Session type |
| | | ' ' | Current session (if session number set to space) |
| | | 'T' | Test frozen session |
| 27 | 1 | | With the UPDT procedure in case of multiple deletion: |
| | | 'N' | Print all transactions, including generated transactions (default option) |
| | | 'O' | Print transactions entered by the user and erroneous generated transactions |
| | | 'E' | Print erroneous transactions only |
| | | | The 2 following fields must be valued for all the extraction procedures which generate update transactions which will modify a Library/session under DSMS control (you can also indicate them on the UPDT '*' line, |
| 40 | 3 | | Product code (3 character-code), |
| 43 | 6 | | Change number (6 character-code, non-significant zeros must be entered), |
| | | | These two codes will appear in the Journal after the execution of UPDT |

| Position | Length | Value | Meaning |
|----------|--------|-------|--|
| 49 | 1 | | Transfer of Entity Lock: |
| | | blank | Replacement of the user code which locks the entity with the user code of the '*' line |
| | | '1' | New entities created from the extracted entities are not locked after the execution of UPDT |
| | | '2' | The user code which locks the entity is kept |
| 50 | 1 | | Transfer of the password on the extraction procedures, on the '*' line of output transactions |
| | | blank | The password is not transferred into the output file, |
| | | '1' | The password is transferred, (Note : for EXTR, the '*' line is transferred into the output file only if you have entered a 'C' in Column 1) |
| 67 | 1 | 'N' | This value is systematically set by the extractors. It indicates that the extracted transactions come from a consistent environment and that updates are always performed with a warning in case of error during a control. Keeping the data consistency triggers, among other things, the inhibition of the lowercase/uppercase conversion and the acceptance of Data Elements formats greater than 999 characters. |

Access Authorizations

An '*' line with a user code and password is required by all procedures.

The Administrator manages the user access authorizations on batch procedures via the Administrator workbench.

Abnormal Endings

Abends may occur during the execution of a batch program. Input-output errors on the system files or on the Database cause a forced abnormal end with a USER ABEND (code 12), accompanied by a message on the SYSOUT file.

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

```

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

```

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

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

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

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

If there is no message and if the type of ABEND is directly related to a problem in the VisualAge Pacbase system programs, contact the VisualAge Pacbase support at IBM. KEEP ALL LISTINGS that may be necessary to analyze the problem.

Performances

Use of the LSR batch system option

The use of LSR is generalized for the files of the Administration Database (GN GR GY GJ GU), the files of the Development Database (AN AR AY AJ), as well as for the OLSD-DBD (SG), Pacbench C/S (SS) and eBusiness (SN) skeletons.

If it is not possible to use the LSR option, delete the 'DD SUBSYS' line and copy its DDNAME on the DSNAME line.

Example :

```
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
```

```
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
```

is replaced by:

```
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
```

Chapter 2. Generation and Printing

GPRT - The Generation/Print Procedure

GPRT - Introduction

The Generation and Printing procedure, GPRT, has a two-fold purpose:

- To print documentation using data contained in the Database, and
- To generate Programs, Screens, eBusiness components, Database descriptions, Data Structures and error messages.

This procedure does not affect the Database. Therefore, it may be executed while the files are open to on-line use.

However, if the generation-print requests submitted on line (+AG) are to be included, the files of the Development Database must be closed. The procedure invalidates the print requests submitted on line, therefore the file must be accessible for update. In this case, the AJ file, which is allocated by default as DUMMY in order to prevent conflicts with the archiving procedure, will have to be allocated (the AJ file is necessary for the *AG command only).

GPRT calls only one program (BVPACB), which is used as a monitor which calls the different programs that make up the procedure.

All the programs that make up the procedure are thus considered as sub-programs of this monitor, with which they communicate via a communication area and specific return codes.

To process all the various user requests, this procedure is broken down into 'sub-chains' whose purpose is to process, in an integrated manner, the preparation of the generation-print requests for the types they manage.

Following the execution of the two general programs that are common to all chains (BVPACA10 and BVPACA20), the sub-chains are activated, if appropriate, in the following order:

- Database Blocks,
- SQL Database Blocks,
- COBOL programs,
- On-line Screens,
- Client Screens,
- Server Screens,

- eBusiness Error Messages,
- Error Messages and Dialog Windowing,
- Personalized Documentation Manager,
- Batch programs,
- Specifications Dictionary.

The files which contain the 'generated source code' (ready to be compiled or to be stored in an Assembler or Source Library) are concatenated into a single physical file that will be used in the following step.

The User Error Message file is updated using the LG-suffixed, and is retrieved into a GL-suffixed file. This file is used to update the User Error Message file. It is used in input to the EMLD or EMUP procedures. In addition, these elements are printed in the IL-suffixed file.

The installed procedure does not provide names for the two versions of this file. Therefore, the names must be specified when these messages are generated.

Volumes are standardly printed in an IN-suffixed file. The GN-suffixed file can also be used (record length = 265) with the 'ASA' skip character in the first position of each record when special print characteristics are needed.

The file containing the elements necessary for the windowing of OLSD applications is coded PAC7GT (record length is 260). Its name must be specified in the generation request.

RPPz Utilities:

With the C9 option, this procedure generates the entities which contain micropatterns.

It generates the micropatterns for the -W lines (types I, E, S, F) which do not come from Macros.

The generation is carried out without line numbers.

The right part (columns 73 to 80) is generated as a blank, except for the function tag lines where lvnn is generated (with nn = level of the function/subfunction).

Execution conditions

The files can remain open, except if the generation-print requests have been submitted on line via the '+AG' command. In this case, the files of the Development Database must be closed.

Abnormal execution

Refer to chapter 'Overview', subchapter 'Abnormal Endings' in the Administrator's Procedures' manual.

GPRT and the SCM module

If the SCM module is available on the site, the generation may create transactions in the QJ file, an archived journal file which contains generated COBOL information such as the Pacbase-constants.

Only the entities defined in an SCM environment and generated from a production session or the current session are recognized to complete QJ.

The QJ transactions can be automatically transferred into the Development Database(s) after the generation, with options specified as parameters in the generation step. The files of the Development Database(s) can remain open.

So the generated entities defined in the SCM Environments are complemented with information related to the last processing of these entities. The status of the entities generated in the current session becomes 'production in wait'.

If errors are found, they are stored in the QJ file. They are printed in output of the ARPM procedure (transactions archiving), and the erroneous transactions are restored in the QJ file in order to be processed again.

To automatically transfer the transactions, saved upon generation, to the Development Database(s), the BVPACB program can receive four parameters separated with commas, as follows:

- the first parameter (SCMMVT) indicates a request for transactions transfer when it is set to "YES" ; the default value is "NO".
- the second parameter (BASE) is used to specify the Development Database code (4-char code) to which the transactions will be transferred ; this parameter is required.
- the third parameter (APPLID) indicates the APPLID name of the corresponding CICS ; the default value is "".
- the fourth parameter (BVP) specifies the prefix of the online programs ; it is required and fixed to 'BVP'.

GPRT - User Input / Results

Input

The GPRT procedure requires the following input:

- a line which identifies the user and the generation-print context,
- one line per generation or print request,
- an optional line ('+AG') which takes into account the requests already submitted on line.

Any other type of transaction is ignored.

Results

There are two types of results:

- A report which lists the requests,
- All the printings requested.

Requests are sorted by user/library and are preceded by a 'banner' (title page).

Note

This procedure does not increment the session number.

GPRT - Generation / Print Commands

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 2 | 1 | | Line code |
| | | 'Z' | Default value |
| 3 | 2 | | Processing sequence order |
| | | | This field is used to specify the sequence in which print requests are processed and printed. |
| 5 | 4 | | GENERATION-PRINT COMMANDS |
| | | | NOTE: Input of the entity code is required or optional depending on the command. The following indicators describe the various options: |
| | | | (A) Required entity code input (Batch column 9). |
| | | | (B) Optional entity code input. If omitted, all the occurrences of the entity type are listed in the user's hierarchical view. field. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | (C) Entity code input not allowed. All occurrences of the entity type are listed in the user's hierarchical view. |
| | | | (D) A blank line may be requested. Type an asterisk in the CONTINUATION OF REQUEST INDICATOR (C) field and press the ENTER key. The options for each command are listed below. This corresponds to batch columns 31 to 80 incl. |
| | | | NOTE: Each command may require additional information. The following list identifies these input fields by code. |
| | | | (1) SEL: _ Limit the list by keyword type: enter 'M' for explicit, 'L' for implicit, or blank for both. In batch mode, enter this value in column 30. See also SELECTION OF KEYWORD TYPE. |
| | | | (2) Same as above plus a following line on which a user may enter one or several keywords. This appears as a continuation line in on-line mode and corresponds to batch columns 31 to 80. |
| | | | (3) FORMAT: _ A format may be specified: enter 'I' for internal, 'E' for input, or 'S' for output. Enter these values in column 17 in batch mode. A blank is also valid and means that the default value is desired. See also TYPE TO SELECT. |
| | | | (4) CCF:_ CCB: The code of the control card in front of program and in back of program, respectively. Enter these codes in columns 19 to 22 in batch mode. These codes must be consistent with the codes displayed on the Dialog Definition screen. |
| | | | (5) CCF:___ CCB:___ The code of the control card in front of program and in front of map, and the code of the control card in back of program and in back of map, respectively. The user can override the default control cards. These codes should be consistent with the values on the Dialog Definition screen. In batch mode, use columns 19 to 22. |
| | | | (6) TYPE: __ The user enters the selected type which should be consistent with the corresponding field on the Definition screen of that entity type. In batch mode enter the type in columns 17 and 18. |
| | | | (7) PRINT DOCUMENT Y CHAP/SUBCHAP AND CODE: _ __ __ Specify the chapter and/or subchapter. Enter 'C' for chapter followed by the chapter code, or 'S' for subchapter followed by the chapter and subchapter codes. In batch mode use columns 23 through 27. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | (8) ENV.:__ (CCF:__ CCB:__) For those sites that are using the SCM option, the environment may be specified. In batch mode enter the environment code in column 17 and the corresponding control cards in columns 19 through 22. |
| | | | THESAURUS |
| | | DCK | (C) A complete Description of Keywords defined in the thesaurus which lists the SYNONYM OR DEFINITION field contents associated with each keyword. |
| | | | Note: This data being specified in Inter-Library only, this command cannot be used with the U1 option. Use the C1 or I1 option which gives the same output. |
| | | LCK | (1) (C) A listing of all keywords defined in the thesaurus, with their synonyms. It includes the number of uses of these keywords in the Database. The information is sequenced by code. |
| | | | TEXTS |
| | | DCT | (A) Description of selected Text. |
| | | | Note: If you enter an asterisk in the ENTITY CODE field, the Descriptions of all Text occurrences are printed, sequenced by code. |
| | | DTT | (B) (6) Descriptions of Text occurrences sequenced by type. |
| | | L*T | List of Texts with their paragraphs titles, sequenced by code. |
| | | LCT | (C) List of Text occurrences sequenced by code. |
| | | LKT | (2) List of Text occurrences whose names and/or explicit Keywords contain the Keyword(s) specified. |
| | | LNT | List of Text occurrences sequenced by name. |
| | | LTT | (6) List of Text occurrences sequenced by type. |
| | | | DOCUMENTS (PDM) |
| | | | Note: DOCUMENT entity = VOLUME entity in the VA Pac character-mode interface. |
| | | DCV | (B) Printing of the Description of the Document whose code is entered in the Entity field. When this code is not entered, the Descriptions of all the Documents are printed, sequenced by code. |
| | | FLV | (C) (D) (4) Flow control for Documents. |
| | | LCV | (C) List of Documents sequenced by code. |
| | | LKV | (C) (2) List of Documents selected according to the keyword(s) entered on the continuation line. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | LNV | (C) (2) List of Documents sequenced by name. |
| | | PCV | (B) (D) (7) Printing of the contents of the Document whose code is entered in the ENTITY CODE field. When this code is not entered, the contents of all the Documents are printed, sequenced by code. For local printing in RTF format, the Document must be generated with the C2 option. Selective Printing is documented in the 'Personalized Documentation Manager' manual, chapter Access Commands, subchapter 'Generation-Printing'. |
| | | | ELEMENTS AND PROPERTIES |
| | | DCE | (B) A complete description of the defined Element(s). The information is sequenced by Element code. |
| | | | Note: to display the assigned text, use print option '2'. |
| | | DFE | (B) A listing of the Element(s) not defined in the Specifications Dictionary, with cross-references. |
| | | LAE | (C) List of Elements sequenced by COBOL name. |
| | | LCE | (B) A list of defined Elements sequenced by Element code. |
| | | LKE | (C) (2) A list of Elements and properties sequenced by keyword. |
| | | LNE | (C) A list of Elements and properties sequenced by name. |
| | | LXE | (C) A list of defined Elements and properties which are not used. |
| | | | DATA STRUCTURES |
| | | DCD | (B) A complete Description of the Data Structure(s). This includes cross-references to Programs and Screens and a list of associated Reports and Segments. The information is sequenced by Data Structure code. |
| | | | Note: To get the associated text use print option '2'. |
| | | FLD | (C) (D) (4) This command is used to specify the job card and end- of-job delimiters: flow control of Data Structures. |
| | | | Use the continuation line to define user parameters on the control cards. |
| | | GCD | (A) Generates a COBOL description (COPY book) of the Data Structure. |
| | | | Upon generation, a Segment can contain up to 9999 Data Elements. An error message is displayed in the generation report if this number is exceeded (for more details on generation, refer to the 'Data Dictionary' manual). |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | C3 : Generation of comments which will be used by VA Pac Connector (an eBusiness tool). |
| | | | C4 : All the calls to the DATA and DATASQ P.I.As. will be ignored |
| | | LCD | (C) A list of Data Structures sequenced by code. |
| | | LED | (A) List the error messages defined for the Data Structure and for each Segment. This list only includes messages that have already been generated. |
| | | LKD | (C) (2) A list of the Data Structures whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LND | (C) (2) A list of the Data Structures sequenced by name. |
| | | LOD | (C) A list of Data Structures sequenced by external name. |
| | | LPD | (C) A list of Data Structures sequenced by Program external name. |
| | | LTD | (C) A list of Data Structures sequenced by type. |
| | | | SEGMENTS AND LOGICAL VIEWS |
| | | DCS | (B) (D: with input of the entity code) (3) |
| | | | You must enter the Data Structure code in the ENTITY field, and the last 2 characters of the Segment code(s) on the continuation line (only one continuation line is possible, i.e. 25 Segments). |
| | | | A complete Description of the Segment(s). This includes cross-references to Programs and Screens for the Data Structure and to all entities for the Segment(s) and a list of associated Reports and Segments. For Segments defined as tables (Pactables function), a list of subschemas and subsystems is printed. |
| | | | Note: To get the associated text for both the Segment and the Data Structure, use print option '2'. |
| | | LCS | (C) List of Segments sequenced by code. |
| | | LKS | (C) (2) List of Segments whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNS | (C) List of Segments sequenced by name. |
| | | | INPUT AIDS |
| | | DCI | (C) A complete description of the Input Aid(s) including a list of uses of the Input Aid(s) in other entities. The information is sequenced by the PIA code. |
| | | LCI | (C) A list of Input Aids sequenced by the PIA code. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | LKI | (C) (2) A list of the Input Aids whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNI | (C) (2) A list of the Input Aids sequenced by name. |
| | | LXI | (C) List of all cross-references (PIA calls) as defined on the PIA Description screen sequenced by the value of this field. |
| | | | DATABASE BLOCKS |
| | | DTB | (B) (6) Description(s) of Database Blocks of the type specified including cross-references to other Blocks and Screens. |
| | | | Note: To get the associated text, use print option '2' |
| | | FLB | (C) (D) (4) (8) This command is used to specify the job card and end- of-job delimiters: Flow control of the Database Block. |
| | | FLS | (C) (D) (4) (8) Same as FLB for Relational/SQL Blocks. |
| | | | Use the continuation line to define user parameters on the control cards. |
| | | GCB | (A) (D) (4) Generate a DDL description of the Database Block specified (including 'DB'-type Blocks for DB2). |
| | | | Use the continuation line to define the user parameters on the control cards. |
| | | | Upon generation, the Segments called in a Block can contain up to 999 Data Elements each. An error message is displayed in the generation report if this number is exceeded. |
| | | GSQ | (A) (D) (4) Generates the SQL DDL for the Relational/SQL Database Block specified. Use the continuation line to define the user parameters on the control cards. |
| | | | Upon generation, the Segments called in a Block can contain up to 999 Data Elements each. An error message is displayed in the generation report if this number is exceeded. |
| | | LCB | (C) List of Database Blocks sequenced by code. |
| | | LEB | (C) List of Database Blocks sequenced by external name. |
| | | LES | (C) List of SQL objects sequenced by external name. |
| | | LKB | (C) (2) A list of the Database Blocks whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNB | (C) (2) A list of Database Blocks sequenced by name. |
| | | LTB | (6) A list of Database Blocks whose Block type have been defined with the specified value. |
| | | LTS | (C) A list of SQL objects sequenced by code. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | * FOLDERS, FOLDER VIEWS, BUSINESS COMPONENTS, * C/S SCREENS (TUI CLIENT COMPONENTS) * SCREENS, DIALOGS. |
| | | DCO | (A) Complete Screen Description including Dialog Complement and uses in other Screens. |
| | | DGC | (A) Complete Description of a Pacbench C/S Screen. |
| | | DGS | (A) Complete Description of a Pacbench C/S Business Component. |
| | | DSO | (A) Description of the selected Screen. |
| | | FGC | (C) (D) (4) (8) Flow control for Pacbench C/S Screens. |
| | | FGE | (C) (D) (4) Flow control for Pacbench C/S error messages. |
| | | FGS | (4) Flow control for Server Component. |
| | | FLE | (C) (D) (4) Flow control for Dialog error messages. |
| | | FLO | (C) (D) (4) (8) Flow control for Screens. |
| | | FME | (4) Flow control for eBusiness Error messages. |
| | | FMS | (4) Flow control for Server. |
| | | FSO | (C) (D) (4) (8) Flow control for source Screen. |
| | | GCO | (A) (D) (5) Generates a COBOL Description of the Screen specified. |
| | | | Upon generation, the Segments called in a Screen can contain up to 999 Data Elements each. An error message is displayed in the generation report if this number is exceeded. |
| | | | Note: If you have specified the 'COBOL formatting option' on the Library, up to 150 lines are possible in a COBOL paragraph (between two periods). |
| | | GGC | (A) (D) (5) Generates a C/S Screen (TUI Client Component). |
| | | | Upon generation, the Segments called in a C/S Screen can contain up to 999 Data Elements each. An error message is displayed in the generation report if this number is exceeded. |
| | | GGs | (A) (D) (5) Generation applicable to Business Component, Communication Monitor, Error Server, Folder. |
| | | | Upon generation, the called Segments can contain up to 999 Data Elements each. An error message is displayed in the generation report if this number is exceeded. |
| | | GEC | (A) (D) Pacbench C/S: |
| | | | C1 : Error messages defined for the Client or Server Dialog and for each component. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | Note: In the 'LANG' field which is displayed after a transmit, you can enter the generation language (EN or FR) of the error messages. If you do not enter any language code, the messages will be generated in your assigned language. If you enter a code other than EN or FR, the messages will be generated in English. |
| | | | C2 : Error messages generated through option 1 plus documentary help messages. |
| | | | C3 : Error messages for the Dialog only. |
| | | GED | (A) (D) |
| | | | C1 : Error messages generated for a Data Structure and for each Segment. |
| | | | Note: In the 'LANG' field which is displayed after a transmit, you can enter the generation language (EN or FR) of the error messages. If you do not enter any language code, the messages will be generated in your assigned language. If you enter a code other than EN or FR, the messages will be generated in English. |
| | | | C2 : Error messages generated through option 1 plus documentary help messages. |
| | | GEO | (A) (D) OLSD Function: |
| | | | C1 : Error messages defined for the Dialog and for each Dialog Screen. |
| | | | Note: In the 'LANG' field which is displayed after a transmit, you can enter the generation language (EN or FR) of the error messages. If you do not enter any language code, the messages will be generated in your assigned language. If you enter a code other than EN or FR, the messages will be generated in English. |
| | | | Note: The header of the generation report displays a summary report of the errors detected during generation and the list of the Screens which have not been generated. |
| | | | C2 : Error messages generated through option 1 plus documentary help messages. |
| | | | C3 : Error messages for the Dialog only. |
| | | | C4 : Generation of a file which contains the data required for the Screen revamping with the Dialog Web Revamping module. If the Screen code includes special characters, an error is generated. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | Note: If a Segment/Screen suffix is entered on the continuation line of one of the preceding commands, error messages are generated/printed only for the selected Segment/Screen. |
| | | GEF | (A) Generation of error messages for a C/S Folder. |
| | | | Note: In the 'LANG' field which is displayed after a transmit, you can enter the generation language (EN or FR) of the error messages. If you do not enter any language code, the messages will be generated in your assigned language. If you enter a code other than EN or FR, the messages will be generated in English. |
| | | GEI | (A) Generation of error messages for INIT/TERM component. |
| | | | Note: In the 'LANG' field which is displayed after a transmit, you can enter the generation language (EN or FR) of the error messages. If you do not enter any language code, the messages will be generated in your assigned language. If you enter a code other than EN or FR, the messages will be generated in English. |
| | | GES | (A) Generation of error messages for a C/S Component. |
| | | | Note: In the 'LANG' field which is displayed after a transmit, you can enter the generation language (EN or FR) of the error messages. If you do not enter any language code, the messages will be generated in your assigned language. If you enter a code other than EN or FR, the messages will be generated in English. |
| | | GSO | (A) Generates source code for the selected Screen. |
| | | GVC | (A) (D) (5) Extract a Proxy object. Applicable to Folder View, Folder and Business Component. |
| | | GMF | (A) Generates a Folder. |
| | | GMI | (A) Generates an INIT/TERM Server. |
| | | GMM | (A) Generates a Communication Monitor. |
| | | GMS | (A) Generates a Server. |
| | | | Upon generation, the Segments called in the Server can contain up to 999 Data Elements each. An error message is displayed in the generation report if this number is exceeded. |
| | | GME | (A) Generates an Error Server. |
| | | LCO | (C) |
| | | | List of Screens sequenced by code. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | LEC | (A) List the error messages defined for the Client Component and for each Client Screen. This list only includes messages that have already been generated. |
| | | LEO | (A) List the error messages defined for the Dialog and for each Screen. This list only includes messages that have already been generated. |
| | | LKO | (C) (2) List of Screens whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNO | (C) List of Screens sequenced by name. |
| | | LOT | (C) List of Screens sequenced by the entered Transaction code. |
| | | LPO | (C) List of C/S Screens sequenced by external program name. |
| | | LSO | (C) List of C/S Screens sequenced by external map name. |
| | | LTO | (C) List of Screens sequenced by type. |
| | | | REPORTS |
| | | DCR | (B) (D: when the entity code has been entered) |
| | | | Note: In the ENTITY field, you must enter a Report code (on 3 characters) to print the Description of one Report, or blanks to print the descriptions of all the Reports. (The commands created with versions earlier than the 3.5 V05 must be manually modified before submitting the print request). |
| | | LCR | (C) List of Reports sequenced by code. |
| | | LTR | (C) List of Reports sequenced by type. |
| | | LKR | (2) A list of the Reports whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNR | (C) List of Reports sequenced by name. |
| | | | PROGRAMS |
| | | DCP | (B) A complete description of Program(s). The information is sequenced by the Program code. |
| | | | Note: To get the associated text, use print option '2'. |
| | | DSP | (A) Description of the selected Program produced by Reverse Engineering. |
| | | FLP | (C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control for Programs. |
| | | | Use the continuation line to define user parameters on the control cards. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | FSP | (C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control for 'reverse engineered' programs. Use the continuation line to define user parameters on the control cards. |
| | | GCP | (A) (D) (4) Generates a COBOL description of the Program specified Use the continuation line to define user parameters on the control cards. Upon generation, the Segments of the Data Structures called in the Program can contain up to 9999 Data Elements. An error message is displayed in the generation report if this number is exceeded. |
| | | | Note: If you have specified the 'COBOL formatting option' on the Library, up to 150 lines are possible in a COBOL paragraph (between two periods). |
| | | GSP | (A) (D) (4) Generate a COBOL description of the 'reverse engineered' Program specified. Use the continuation line to define user parameters on the control cards. |
| | | LCP | (C) List of Programs sequenced by program code. Note: To get keywords, use print option '2'. |
| | | LEP | (C) List of Programs sequenced by external name. |
| | | LKP | (2) A list of the Programs whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNP | (2) List of Programs sequenced by name. |
| | | LTP | (C) List of Programs sequenced by type. |
| | | | METHOD ENTITIES |
| | | DCM | (A) A complete Description of the Method entity as specified. |
| | | DCMC | (C) A complete Description of Method Functional Integrity Constraint(s). |
| | | DCMO | (C) A complete Description of Method Object(s). |
| | | DCMR | (C) A complete Description of Method Relation(s). |
| | | LCMC | (C) List of Method Functional Integrity Constraints sequenced by F.I.C. code. |
| | | LCMO | (C) List of Method Objects sequenced by Object code. |
| | | LCMP | (C) List of properties sequenced by Property code. |
| | | LCMR | (C) List of Method Relations with their Functional Integrity Constraints, sequenced by Relation code. |
| | | LKM | (C) (2) A list of the Method entities whose names and/or explicit keywords contain the keyword(s) specified. |
| | | | META-ENTITIES |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | DCF | (B) A complete Definition and Description of the Meta-Entity entered in the ENTITY field. If no code is specified, all Meta-Entities are listed. The information is sequenced by code. |
| | | DCQ | (B) A complete Definition and Description of the User Relations entered in the ENTITY field. If no code is specified, all User Relations are listed. The information is sequenced by code. |
| | | DCY | (B) A complete Definition and Description of the Extended User Entity entered in the ENTITY field. If no code is specified, all Extended User Entities are listed. The information is sequenced by code. |
| | | DC\$ | (B) A complete Definition and Description of the User Entity entered in the ENTITY field (the following form is required: DC\$xx, where xx corresponds to the type of entity call). |
| | | LCF | (C) List of Meta-Entities sequenced by code. |
| | | LCQ | (C) List of User Relations sequenced by code. |
| | | LCY | (A) List of Extended User Entities sequenced by code. |
| | | LC\$ | (A) List of User Entities sequenced by type and code (LC\$xx, xx being the type of entity call). |
| | | LKF | (2) (C) A list of the Meta-Entities whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LKQ | (2) (C) A list of the User Entities Relations whose names and/or explicit keywords contain the keyword(s) specified. |
| | | | Note: For all printing by keyword, you can specify the TYPE OF SELECTION (BLANK, L or M) on the print line. Keywords are indicated on the continuation line sent back. |
| | | LKY | (2) (A) A list of the Extended User-Entities whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LK\$ | (2) (A) A list of the User Entities whose names and/or explicit keywords contain the keyword(s) specified. |
| | | LNF | (C) A list of the Meta-Entities sequenced by name. |
| | | LNQ | (C) A list of the User Relations sequenced by name. |
| | | LN Y | (A) A list of Extended User-Entities sequenced by name. |
| | | LN\$ | (A) A list of the User Entities sequenced by name. |
| | | | SHIFT TO UPPER-CASE |
| | | UPC | This command allows for the automatic transformation of lower-case letters into upper-case letters in the printed output of the GPRT procedure. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | When the UPC command is entered, the following line is displayed: |
| | | | SHIFT TO UPPERCASE MANUAL:_ DOC:_ ERROR MESS: |
| | | | The VA Pac user must specify to which type of GPRT output the UPC command will apply (even when only one GPRT command is validated). |
| | | | In order to do this, the value '1' must be entered in one of the three fields displayed above: in the MANUAL field for Volumes (V); in the DOC field for entity related commands; in the ERROR MESS field for the generation of error messages. |
| | | | Note: This also allows the selective implementation of the UPC command when the execution of several GPRT jobs is requested and the SHIFT TO UPPERCASE must not apply to all of them, in which case the corresponding field(s) must be left blank. |
| | | GUT | (A) Generate User Command. |
| | | | A GUT command is constituted of a first comment line and of one to five continuation lines which contain the data to be sent to the generation flow. There is no control on this command code. |
| | | | First line: The entity code is optional. The label is made up of two parts which can be modified. The first part (on 25 characters) is initialized to 'USER GENERATION' and the second part (on 11 characters) must be completed by the user. Only this second part is sent to the generation procedure and is displayed in the PAC7ID report. |
| | | | Continuation lines: five continuation lines are authorized. They are simply sent to the generation flow. They are found in the PAC7ID report and in the PAC7JC intermediate file. To process this type of request, you must then adapt your generation flow. |
| | | | Note: a GUT command without any continuation line nevertheless generates a line in the generation flow but with a space in the command. |
| | | | PAF TABLES FOR METHODOLOGY ENTITIES |
| | | PCM | Description of PAF Tables for entities specific to a methodology. This command is necessarily followed by a Methodology code. |
| 9 | 6 | | ENTITY CODE |
| | | | This field is displayed with the label 'ENTITY' on screen format options '1' and '2' of the GP screen. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | When required, the user enters the entity code which corresponds to the COMMAND FOR PRINT REQUEST. |
| | | | 'PCM' COMMAND: In this field, you enter the code of the selected Methodology: |
| | | M | Merise |
| | | D | YSM |
| | | A | SSADM |
| | | O | OMT |
| | | F | IFW |
| 15 | 1 | | Library selection indicator |
| | | | Used to select the libraries from which the entities are to be generated and/or printed. |
| | | C | Selected library and higher level libraries. In case of duplicates, the lines from the lower level library are taken into account. |
| 16 | 1 | | PRINT OPTION |
| | | | In this field, you specify print options. |
| | | | There are 5 options numbered from 1 to 4, and 9. The default option is 1. |
| | | | Each option corresponds to presentation variants of lines to be printed, e.g. printing of additional information (with or without keywords, programs with or without associated texts, ...). |
| | | | The detail of each print option is given for each entity in the corresponding reference Manuals. |
| | | | Value 9 is reserved for RPPz. |
| 17 | 2 | | Generation criteria |
| | | | Used to enter the language code for the GEx generation-print commands. |
| 19 | 1 | | Control cards in front of programs |
| | | | Enter the one-character code that identifies the job card to be inserted before the generated program. |
| | | | Default: Code entered on the Library Definition Screen |
| 20 | 1 | | CONTROL CARDS BEFORE MAP |
| | | | Screen and C/S Screen entities |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | Option code that identifies the job card to be inserted before each generated Screen or C/S Screen map. |
| | | \$ | No generation of map. |
| | | | NOTE: This field is not used in a Pacbench C/S development with specification of Folder. |
| | | | Business Component / single-view (with no specification of a Folder). |
| | | | Option code which selects the JCL lines to be inserted before the generated Services Manager. The value '\$' is used to disable the generation of the Services Manager and to enable the Business Component to be generated. |
| 21 | 1 | | CONTROL CARDS IN BACK OF PROGRAMS |
| | | | Enter the one-character code that identifies the job card to be inserted after the generated program. |
| | | | Default: Code entered on the Library Definition Screen |
| 22 | 1 | | CONTROL CARDS AFTER MAP |
| | | | Screen and C/S Screen entities: |
| | | | Option code that identifies the job card to be inserted after each generated Screen or Screen c/s map. |
| | | \$ | No generation of map. |
| | | | NOTE: This field is not used in a Pacbench C/S development with the specification of Folder. |
| | | | Business Component / single-view (with no specification of Folder): |
| | | | Option code which selects the JCL lines to be inserted after the Services Manager generated. |
| 23 | 1 | | DOCUMENT SELECTIVE PRINT REQUEST |
| | | | Field displayed with PCV command only. |
| | | blank | Print the whole Document (default value) |
| | | C or 1 | Print the selected chapter or level-1 section, respectively. Field used jointly with next field. |
| | | S or 2 | Print the selected subchapter or level-2 section (included in the level-1 section indicated in the following field), respectively. Field used jointly with next two fields. |
| 24 | 2 | | Level-1 Section # / Chapter Code |
| | | C | The value 'ZZ' is not authorized. CH/ |
| 26 | 2 | | Level-2 Section # / Subchapter Code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | C | SC/ |
| 30 | 1 | | SELECTION OF KEYWORD TYPE |
| | | blank | Selection on both implicit and explicit keywords. |
| | | L | Selection on implicit keywords only. |
| | | M | Selection on explicit keywords only. |
| 31 | 50 | | Label continuation |

GPRT - Processing of Job Streams

When there has been a generation of source code to be compiled and when the return code of the preceding step is less than 8, the generated stream must be processed: compilation, assembly and link-edit of the product source code.

Two standard options are available for processing this job stream upon installation:

- Submission of the job stream by sending the generated program flow to the INTERNAL READER: GPRT procedure or GPRP procedure (if the PAF pre-processor is used),
- Storage of the generated source code in a symbolic library. All compiles executed after this are the responsibility of each user (GPRU procedure).

These two options are not the only possibilities. Each site can set up its own procedure for job stream processing.

Whatever solution is adopted, the different generated source codes must be preceded, or followed by, a set of optional control cards.

GPRT - Description of Steps

Input recognition: PTU001

Preparation of the KSDS work-file DEFINE: PRMSYS

This program is used to parameterize the KSDS work file name. The file DELETE/DEFINE is found in the DFSYSPAF member of the SY parameters PDS. Its Dsname is suffixed with the &USER parameter. The PRMSYS program sets this parameter to the value entered at execution time. The purpose is to allocate different work files from one execution to another and to avoid any conflicts when allocating work files in case of simultaneous executions.

| Code | Physical name | Type | Label |
|--------|--------------------------|--------|--------------------------------|
| PACRIN | &INDSN..BVPSY(DFSYSYPAF) | Input | File DEFINE skeleton |
| PACROU | &&DFSYSYPAF | Output | File DEFINE (SYSIN for IDCAMS) |

Definition of KSDS work file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DFSYSYPAF | Input | File DEFINE |

Check of VSAM files: IDCAMS

Generation and printing: PACB

The generated documentation depends on the generation-print requests taken into account. Therefore, the volume of the generated documentation and of the temporary files is extremely variable.

Banners at the beginning and at the end of user documentation, which display the user code, facilitate the identification of their authors.

All the Programs, Screens, Database Blocks that have been generated are retrieved in a single file: DSN=&INDUV..&BASE..GN&USER, which will be used in the next step.

Some programs called by the monitor can generate specific return codes:

- BVPACA10 (Retrieval of Transactions) :
 - 0 : OK
 - 2 : OK with presence of the '+AG' command
 - 8 : No request.
 In this case, the procedure stops running.
- BVPACB31 (SQL generation):
 - 8 : Error detected during generation.
- Extractors or generators (30 or 40):
 - 0 : OK - No generation
 - 4 : OK - Generation
 - Other : Errors
- BVPACW10 (configuration management support)
 - 0 : OK
 - 2 : No processing

4 : at least one parameterizing error detected.

8 : at least one context error detected.

This step sends a general return code.

| Code | Label |
|------|--|
| 4 | OK with generation of source code |
| 6 | OK with generation of source code and Personalized Documentation or error messages |
| 8 | OK with generation of Personalized Documentation or of error messages |
| 10 | OK without generation |
| 12 | Input-Output error |
| 16 | Sort error |

Creation of generator output file: IEBGENER

This step is executed only when the procedure used is GPRT.

The created file is DSN=*.PAC.PAC7OB

Update of generator output file: IEBUPDTE

This step is executed only when the procedure used is GPRU.

| Code | Physical name | Type | Label |
|--------|---------------|--------|-------|
| SYSIN | *.PAC.PAC7OB | Input | |
| SYSUT2 | &&BIBS | Output | |

Preparation of the KSDS work-file DELETE: PRMSYS

This program's purpose is to parameterize the work file name. The DELETE of the file is found in the DFSYSPAF member of the SY parameters PDS.

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--------------------------------|
| PACRIN | &INDSN..BVPSY(DLSYSPAF) | Input | File DELETE skeleton |
| PACROU | DSN=&&DLSYSPAF | Output | File DELETE (SYSIN for IDCAMS) |

Deletion of the KSDS work-file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DLSYSPAF | Input | File DELETE |

GPRT - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - GENERATION (IN INTERNAL READER) AND PRINTING -
/**
/** -----
/**
/** IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST
/** CONTAIN THE JCL REQUIRED TO COMPILE THEM,
/** USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND
/** THE BEFORE/AFTER PROGRAM OPTIONS.
/**
/** THE GENERATION AND PRINTING PROCEDURE, GPRT, HAS A
/** TWO-FOLD PURPOSE:
/**   . TO PRINT DOCUMENTATION USING DATA CONTAINED IN THE
/**     DATABASE, AND
/**   . TO GENERATE PROGRAMS, SCREENS, DATABASE
/**     DESCRIPTIONS DATA STRUCTURES, AND ERROR MESSAGES.
/**
/** -----
/**
/**/BVPGPRT PROC BASE=$BASE,                CODE OF VAPAC DATABASE
/**      INDSV='$INDSV',                    INDEX OF VSAM SYSTEM FILES
/**      INDSN='$INDSN',                    INDEX OF NON-VSAM SYSTEM FILES
/**      INDUV='$INDUV',                    INDEX OF VSAM USER FILES
/**      INDUN='$INDUN',                    INDEX OF USER NON VSAM FILES
/***:   VSAMCAT='$VCAT',                    USER VSAM CATALOG
/***:   SYSTCAT='$SCAT',                    SYSTEM VSAM CATALOG
/**      STEPLIB='$HLQ..SBVPMBR8',          LIBRARY OF LOAD-MODULES
/**      SORTLIB='$BIBT',                   SORT LIBRARY
/**      COPIES=1,                          NUMBER OF REPORT COPIES
/**      NBMAN=1,                            NUMBER OF COPIES OF VOLUME (PDM)
/**      OUT=$OUT,                          UTILITIES AND ERRORS OUTPUT CLASS
/**      OUTL=$OUT,                          OUTPUT CLASS OF REPORTS
/**      VOL='$SER=$VOLUN',                  VOLUME OF GENERATION FILE
/**      UNIT=$UNITUN,                       DISK UNIT OF GENERATION FILE
/**      UWK=$UWK,                           WORK UNIT
/**      USER=,                              USER CODE
/**      SCMMVT='NO',                        SCM TRANSACTION
/**      APPLID=' ',                          CICS APPLID
/**      LSR='BLSR',                          LSR BATCH SYSTEM NAME
/**      SPAGN='(TRK,(100,10))',             GENERATION FILE SPACE
/**      SPAEX='(TRK,(50,10),RLSE)',         SPACE OF PRINTING FILES
/**      SPAWK='(TRK,(50,20))',             WORK FILE SPACE
/**      SPAMB='(TRK,(5,1),RLSE)',          REQUEST FILE SPACE
/**      SPAMAN='(TRK,(50,10),RLSE)',        VOLUME (PDM) FILE SPACE
/**      SPAWIN='(TRK,(50,10),RLSE)',        WINDOWING FILE SPACE
/**      SPAIDX='(TRK,(50,10),RLSE)' INDEX OF VOLUME FILE SPACE

```



```

//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&GPRTMB,DISP=(,PASS),UNIT=&UWK,
// DCB=BLKSIZE=3440,SPACE=&SPAMB
//CARTE DD DDNAME=SYSIN
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYSYPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSYPAF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSYPAF DD DSN=&INDUV..SYSYPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PAC7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFLB),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFQJ),DISP=SHR
//PAC EXEC PGM=BVPACB,REGION=0K,
// PARM='&SCMMVT,&BASE,&APPLID,BVP'
//*-----
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR

```

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//* DD DSN=$DFHEXC,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//AELSR DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PAC7AJ DD DUMMY
//*AJLSR DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//*PAC7AJ DD SUBSYS=(&LSR,'DDNAME=AJLSR')
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR')
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PACG GK DD DSN=&INDSV..BVP GK,DISP=SHR
//GNLSR DD DSN=&INDSV..BVP GN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR')
//GRLSR DD DSN=&INDSV..BVP GR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR')
//PACGGU DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7DG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=12560
//PAC7EB DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EE DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EI DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7EP DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EQ DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7ER DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EW DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7GB DD DSN=&&PAC7GB,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GD DD DSN=&&PAC7GD,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GE DD DSN=&&PAC7GE,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GF DD DSN=&&PAC7GF,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GG DD DSN=&&PAC7GG,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GI DD DSN=&&PAC7GI,UNIT=&UWK,DCB=BLKSIZE=3440,
// SPACE=(TRK,(10,5),RLSE),DISP=(,DELETE)
//PAC7GK DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GL DD DSN=&&PAC7GL,UNIT=&UWK,
// SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=1800

```

```

//PAC7GM DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GN DD DSN=&&PAC7GN,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=265,BLKSIZE=6095),
// SPACE=&SPAMAN
//PAC7GO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=4140
//PAC7GP DD DSN=&&PAC7GP,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GQ DD DSN=&&PAC7GQ,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GR DD DSN=&&PAC7GR,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//PAC7GT DD DSN=&&PAC7GT,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=260,BLKSIZE=7800),
// SPACE=&SPAWIN
//PAC7GV DD DSN=&&PAC7GV,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7G6 DD DSN=&&PAC7G6,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=368,BLKSIZE=27232),
// SPACE=&SPAMAN
//PAC7DB DD SYSOUT=&OUTL
//PAC7IA DD SYSOUT=&OUTL
//PAC7ID DD SYSOUT=&OUTL,COPIES=&COPIES
//PAC7IK DD SYSOUT=&OUTL
//PAC7IL DD SYSOUT=&OUTL
//PAC7IN DD SYSOUT=&OUTL,COPIES=&NBMAN
//PAC7IO DD SYSOUT=&OUTL
//PAC7IW DD SYSOUT=&OUTL
//PAC7JG DD DSN=&&PAC7JG,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440),
// SPACE=&SPAMB
//PAC7KB DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KD DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KE DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KF DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KM DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7KP DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KQ DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KR DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KS DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KU DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7LG DD DUMMY,DCB=BLKSIZE=90
//PAC7LI DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3440
//PAC7LK DD DUMMY,DCB=BLKSIZE=100
//PAC7LM DD DUMMY,DCB=BLKSIZE=100
//PAC7ME DD DSN=&&GPRTMB,DISP=(OLD,DELETE,DELETE)
//PAC7MG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3060
//PAC7MV DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=6800

```

```

//PAC70B DD DSN=&INDUN..&BASE..GN&USER,
//        DCB=BLKSIZE=12560,
//        UNIT=&UNIT,
//        VOL=&VOL,
//        SPACE=&SPAGN,DISP=(,PASS,DELETE)
//PAC70D DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70E DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70F DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70G DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70P DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70Q DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70R DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//PAC70V DD DSN=*.PAC70B,
//        VOL=REF=*.PAC70B,
//        DISP=(MOD,,DELETE)
//QJLSR DD DSN=&INDSV..BVPQJ,DISP=SHR
//PAC7QJ DD SUBSYS=(&LSR,'DDNAME=QJLSR','BUFND=10')
//PAC7SC DD DSN=&INDSV..BVPSC,DISP=SHR
//SGLSR DD DSN=&INDSV..BVPSG,DISP=SHR
//PAC7SG DD SUBSYS=(&LSR,'DDNAME=SGLSR','BUFND=10')
//SNLSR DD DSN=&INDSV..BVPSN,DISP=SHR
//PAC7SN DD SUBSYS=(&LSR,'DDNAME=SNLSR','BUFND=10')
//PAC7S0 DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),
//        DCB=(RECFM=FB,LRECL=576,BLKSIZE=27648)
//PAC7SR DD DSN=&INDSV..BVPSR,DISP=SHR
//SSLSR DD DSN=&INDSV..BVPSS,DISP=SHR
//PAC7SS DD SUBSYS=(&LSR,'DDNAME=SSLSR','BUFND=10')
//PAC7WA DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=25200
//PAC7WC DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W1 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W2 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W3 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W4 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27180
//PAC7W6 DD UNIT=&UWK,SPACE=&SPAMAN,
//        DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W7 DD UNIT=&UWK,SPACE=&SPAWK,
//        DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W8 DD UNIT=&UWK,SPACE=&SPAMAN,
//        DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W9 DD UNIT=&UWK,SPACE=&SPAIDX,
//        DCB=(RECFM=FB,LRECL=55,BLKSIZE=12595)

```

```

//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAFLSR DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//SYSPAF DD SUBSYS=(&LSR,'DDNAME=PAFLSR')
//SYSUDUMP DD SYSOUT=&OUT
//RDR EXEC PGM=IEBGENER,COND=(8,LE,PAC)
//*-----
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DUMMY,DCB=BLKSIZE=80
//SYSUT1 DD DSN=*.PAC.PAC70B,
// VOL=REF=*.PAC.PAC70B,
// DISP=(OLD,DELETE,DELETE)
//SYSUT2 DD SYSOUT=(&OUT,INTRDR)
//DEL EXEC PGM=IEFBR14,COND=(8,GT,PAC)
//*-----
//PAC7GN DD DSN=*.PAC.PAC70B,
// VOL=REF=*.PAC.PAC70B,
// DISP=(OLD,DELETE,DELETE)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)

```

GPRU - Execution JCL

```

/* -----
/* VISUALAGE PACBASE
/*
/* -----
/* - GENERATION (INTO SOURCE LIBRARY) AND PRINTING -
/*
/* -----
/* IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST
/* CONTAIN AN "./ ADD NAME=SOURCE-CODE" CARD PER GENERATED
/* ENTITY USING THE BEFORE/AFTER PROGRAM OPTIONS.
/*
/* -----
/*
//BVPGPRU PROC BASE=$BASE, CODE OF VAPAC DATABASE
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES

```

```

//*:      VSAMCAT='$VCAT',                USER VSAM CATALOG
//*:      SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8',        LIBRARY OF LOAD-MODULES
//      SORTLIB='$BIBT',                SORT LIBRARY
//      COPIES=1,                        NUMBER OF REPORT COPIES
//      NBMAN=1,                          NUMBER OF COPIES OF VOLUME (PDM)
//      OUT=$OUT,                          UTILITIES AND ERRORS OUTPUT CLASS
//      OUTL=$OUT,                          OUTPUT CLASS OF REPORTS
//      VOL='SER=$VOLUN',                  VOLUME OF GENERATION FILE
//      UNIT=$UNITUN,                      DISK UNIT OF GENERATION FILE
//      UWK=$UWK,                          WORK UNIT
//      USER=,                              USER CODE
//      SCMMVT='NO',                        SCM TRANSACTION
//      APPLID='',                          CICS APPLID
//      LSR='BLSR',                          LSR BATCH SYSTEM NAME
//      SPAGN='(TRK,(100,10))',            GENERATION FILE SPACE
//      SPAEX='(TRK,(50,10),RLSE)',        SPACE OF PRINTING FILES
//      SPAWK='(TRK,(50,20))',            WORK FILE SPACE
//      SPAMB='(TRK,(5,1),RLSE)',          REQUEST FILE SPACE
//      SPAMAN='(TRK,(50,10),RLSE)',       VOLUME (PDM) FILE SPACE
//      SPAWIN='(TRK,(50,10),RLSE)',       WINDOWING FILE SPACE
//      SPAIDX='(TRK,(50,10),RLSE)',       INDEX OF VOL. FILE SPACE
//      SPABI='(TRK,(200,10,10))'         SYMBOLIC LIBRARY SPACE
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&GPRTMB,DISP=(,PASS),UNIT=&UWK,
//      DCB=BLKSIZE=3440,SPACE=&SPAMB
//CARTE DD DDNAME=SYSIN
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYSYPAF,DISP=(,PASS),SPACE=(TRK,1),
//      UNIT=&UWK,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSYPAF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSYPAF DD DSN=&INDUV..SYSYPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR

```

```

//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PAC7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFLB),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFQJ),DISP=SHR
//PAC EXEC PGM=BVPACB,REGION=0K,
// PARM='&SCMMVT,&BASE,&APPLID,BVP'
//*-----
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//* DD DSN=$DFHEXC,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//AELSR DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PAC7AJ DD DUMMY
//*AJLSR DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//*PAC7AJ DD SUBSYS=(&LSR,'DDNAME=AJLSR')
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR')
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PACGGK DD DSN=&INDSV..BVP GK,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR')
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7DG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=12560
//PAC7EB DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EE DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EI DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7EP DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EQ DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820

```

```

//PAC7ER DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EW DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7GB DD DSN=&&PAC7GB,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GD DD DSN=&&PAC7GD,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GE DD DSN=&&PAC7GE,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GF DD DSN=&&PAC7GF,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GG DD DSN=&&PAC7GG,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GI DD DSN=&&PAC7GI,UNIT=&UWK,DCB=BLKSIZE=3440,
// SPACE=(TRK,(10,5),RLSE),DISP=(,DELETE)
//PAC7GK DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GL DD DSN=&&PAC7GL,UNIT=&UWK,
// SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=1800
//PAC7GM DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GN DD DSN=&&PAC7GN,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=265,BLKSIZE=6095),
// SPACE=&SPAMAN
//PAC7GO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=4140
//PAC7GP DD DSN=&&PAC7GP,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GQ DD DSN=&&PAC7GQ,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GR DD DSN=&&PAC7GR,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//PAC7GT DD DSN=&&PAC7GT,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=260,BLKSIZE=7800),
// SPACE=&SPAWIN
//PAC7GV DD DSN=&&PAC7GV,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7G6 DD DSN=&&PAC7G6,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=368,BLKSIZE=27232),
// SPACE=&SPAMAN
//PAC7DB DD SYSOUT=&OUTL
//PAC7IA DD SYSOUT=&OUTL
//PAC7ID DD SYSOUT=&OUTL,COPIES=&COPIES
//PAC7IK DD SYSOUT=&OUTL
//PAC7IL DD SYSOUT=&OUTL
//PAC7IN DD SYSOUT=&OUTL,COPIES=&NBMAN
//PAC7IO DD SYSOUT=&OUTL
//PAC7IW DD SYSOUT=&OUTL

```



```

//PAC7JG DD DSN=&&PAC7JG,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440),
//          SPACE=&SPAMB
//PAC7KB DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KD DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KE DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KF DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KM DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7KP DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KQ DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KR DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KS DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KU DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7LG DD DUMMY,DCB=BLKSIZE=90
//PAC7LI DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3440
//PAC7LK DD DUMMY,DCB=BLKSIZE=100
//PAC7LM DD DUMMY,DCB=BLKSIZE=100
//PAC7ME DD DSN=&&GPRTMB,DISP=(OLD,DELETE,DELETE)
//PAC7MG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3060
//PAC7MV DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=6800
//PAC7OB DD DSN=&INDUN..&BASE..GN&USER,
//          DCB=BLKSIZE=12560,
//          UNIT=&UNIT,
//          VOL=&VOL,
//          SPACE=&SPAGN,DISP=(,PASS,DELETE)
//PAC7OD DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OE DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OF DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OG DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OP DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OQ DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OR DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OV DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//QJLSR DD DSN=&INDSV..BVPQJ,DISP=SHR
//PAC7QJ DD SUBSYS=(&LSR,'DDNAME=QJLSR','BUFND=10')
//PAC7SC DD DSN=&INDSV..BVPSC,DISP=SHR

```

```

//SGLSR DD DSN=&INDSV..BVPSG,DISP=SHR
//PAC7SG DD SUBSYS=(&LSR,'DDNAME=SGLSR','BUFND=10')
//SNLSR DD DSN=&INDSV..BVPSN,DISP=SHR
//PAC7SN DD SUBSYS=(&LSR,'DDNAME=SNLSR','BUFND=10')
//PAC7SO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),
// DCB=(RECFM=FB,LRECL=576,BLKSIZE=27648)
//PAC7SR DD DSN=&INDSV..BVPSR,DISP=SHR
//SSLSR DD DSN=&INDSV..BVPSR,DISP=SHR
//PAC7SS DD SUBSYS=(&LSR,'DDNAME=SSLSR','BUFND=10')
//PAC7WA DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=25200
//PAC7WC DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W1 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W2 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W3 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W4 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27180
//PAC7W6 DD UNIT=&UWK,SPACE=&SPAMAN,
// DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W7 DD UNIT=&UWK,SPACE=&SPAWK,
// DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W8 DD UNIT=&UWK,SPACE=&SPAMAN,
// DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W9 DD UNIT=&UWK,SPACE=&SPAIDX,
// DCB=(RECFM=FB,LRECL=55,BLKSIZE=12595)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAFLSR DD DSN=&INDUV..SYSAPF.&USER,DISP=SHR
//SYSAPF DD SUBSYS=(&LSR,'DDNAME=PAFLSR')
//SYSUDUMP DD SYSOUT=&OUT
//UPD EXEC PGM=IEBUPDTE,COND=(8,LE,PAC),PARM=NEW
//*-----
//SYSIN DD DSN=*.PAC.PAC70B,
// VOL=REF=*.PAC.PAC70B,
// DISP=(OLD,DELETE,DELETE)
//SYSPRINT DD DUMMY
//SYSUT2 DD DSN=&&BIBS,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPABI,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//DEL EXEC PGM=IEFBR14,COND=(8,GT,PAC)
//*-----
//PAC7GN DD DSN=*.PAC.PAC70B,
// VOL=REF=*.PAC.PAC70B,
// DISP=(OLD,DELETE,DELETE)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----

```

```

//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)

```

GPRP - Execution JCL

```

//* -----
//*      VISUALAGE PACBASE
//*
//* -----
//*      - GENERATION AND PRINTING WITH PAF PRE-PROCESSING -
//*      (INTO INTERNAL READER)
//* -----
//*
//* IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST
//* CONTAIN THE JCL REQUIRED TO COMPILE THEM,
//* USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND
//* THE BEFORE/AFTER PROGRAM OPTIONS.
//*
//* -----
//*
//BVPGPRP PROC BASE=$BASE,                CODE OF VAPAC DATABASE
//      INDSV='$INDSV',                    INDEX OF VSAM SYSTEM FILES
//      INDSN='$INDSN',                    INDEX OF NON-VSAM SYSTEM FILES
//      INDUV='$INDUV',                    INDEX OF VSAM USER FILES
//      INDUN='$INDUN',                    INDEX OF USER NON VSAM FILES
//*:    VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:    SYSCAT='$SCAT',                      SYSTEM VSAM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8',          LIBRARY OF LOAD-MODULES
//      SORTLIB='$BIBT',                    SORT LIBRARY
//      COPIES=1,                           NUMBER OF REPORT COPIES
//      NBMAN=1,                             NUMBER OF COPIES OF VOLUME (PDM)
//      OUT=$OUT,                             UTILITIES AND ERRORS OUTPUT CLASS
//      OUTL=$OUT,                             OUTPUT CLASS OF REPORTS
//      VOL='SER=$VOLUN',                      VOLUME OF GENERATION FILE
//      UNIT=$UNITUN,                          DISK UNIT OF GENERATION FILE
//      UWK=$UWK,                               WORK UNIT
//      USER=,                                 USER CODE
//      SCMMVT='NO',                           SCM TRANSACTION
//      APPLID='',                             CICS APPLID
//      LSR='BLSR',                             LSR BATCH SYSTEM NAME
//      SPAGN='(TRK,(100,10))',                GENERATION FILE SPACE
//      SPAEX='(TRK,(50,10),RLSE)',            SPACE OF PRINTING FILES
//      SPAWK='(TRK,(50,20))',                WORK FILE SPACE
//      SPAMB='(TRK,(5,1),RLSE)',              REQUEST FILE SPACE
//      SPAMAN='(TRK,(50,10),RLSE)',           VOLUME (PDM) FILE SPACE
//      SPAWIN='(TRK,(50,10),RLSE)',          WINDOWING FILE SPACE
//      SPAIDX='(TRK,(50,10),RLSE)' INDEX OF VOLUME FILE SPACE
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&GPRTMB,DISP=(,PASS),UNIT=&UWK,
//      DCB=BLKSIZE=3440,SPACE=&SPAMB
//CARTE DD DDNAME=SYSIN

```

```

//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYSYPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSYPAF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSYPAF DD DSN=&INDUV..SYSYPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PAC7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFLB),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFQJ),DISP=SHR
//PAC EXEC PGM=BVPACB,REGION=0K,
// PARM='&SCMMVT,&BASE,&APPLID,BVP'
//*-----
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//* DD DSN=$DFHEXC,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//AELSR DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')

```

```

//PAC7AJ DD DUMMY
//*AJLSR DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AJ DD SUBSYS=(&LSR,'DDNAME=AJLSR')
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR')
//PAC7LB DD DSN=&INDSV..BVLB,DISP=SHR
//PACGGK DD DSN=&INDSV..BVPGK,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR')
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7DG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=12560
//PAC7EB DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EE DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EI DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7EP DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EQ DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7ER DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EW DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7GB DD DSN=&&PAC7GB,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GD DD DSN=&&PAC7GD,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GE DD DSN=&&PAC7GE,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GF DD DSN=&&PAC7GF,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GG DD DSN=&&PAC7GG,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GI DD DSN=&&PAC7GI,UNIT=&UWK,DCB=BLKSIZE=3440,
// SPACE=(TRK,(10,5),RLSE),DISP=(,DELETE)
//PAC7GK DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GL DD DSN=&&PAC7GL,UNIT=&UWK,
// SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=1800
//PAC7GM DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GN DD DSN=&&PAC7GN,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=265,BLKSIZE=6095),
// SPACE=&SPAMAN
//PAC7GO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=4140
//PAC7GP DD DSN=&&PAC7GP,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)

```

```

//PAC7GQ DD DSN=&&PAC7GQ,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7GR DD DSN=&&PAC7GR,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//PAC7GT DD DSN=&&PAC7GT,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=260,BLKSIZE=7800),
//        SPACE=&SPAWIN
//PAC7GV DD DSN=&&PAC7GV,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7G6 DD DSN=&&PAC7G6,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=368,BLKSIZE=27232),
//        SPACE=&SPAMAN
//PAC7DB DD SYSOUT=&OUTL
//PAC7IA DD SYSOUT=&OUTL
//PAC7ID DD SYSOUT=&OUTL,COPIES=&COPIES
//PAC7IK DD SYSOUT=&OUTL
//PAC7IL DD SYSOUT=&OUTL
//PAC7IN DD SYSOUT=&OUTL,COPIES=&NBMAN
//PAC7IO DD SYSOUT=&OUTL
//PAC7IW DD SYSOUT=&OUTL
//PAC7JG DD DSN=&&PAC7JG,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440),
//        SPACE=&SPAMB
//PAC7KB DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KD DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KE DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KF DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KM DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7KP DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KQ DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KR DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KS DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KU DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7LG DD DUMMY,DCB=BLKSIZE=90
//PAC7LI DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3440
//PAC7LK DD DUMMY,DCB=BLKSIZE=100
//PAC7LM DD DUMMY,DCB=BLKSIZE=100
//PAC7ME DD DSN=&&GPRTMB,DISP=(OLD,DELETE,DELETE)
//PAC7MG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3060
//PAC7MV DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=6800
//PAC7OB DD DSN=&INDUN..&BASE..GN&USER,
//        DCB=BLKSIZE=12560,
//        UNIT=&UNIT,
//        VOL=&VOL,
//        SPACE=&SPAGN,DISP=(,PASS,DELETE)
//PAC7OD DD DSN=*.PAC7OB,
//        VOL=REF=*.PAC7OB,
//        DISP=(MOD,,DELETE)

```

```

//PAC70E DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70F DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70G DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70P DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70Q DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70R DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70V DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//QJLSR DD DSN=&INDSV..BVPQJ,DISP=SHR
//PAC70J DD SUBSYS=(&LSR,'DDNAME=QJLSR','BUFND=10')
//PAC70S DD DSN=&INDSV..BVPSC,DISP=SHR
//SGLSR DD DSN=&INDSV..BVPSG,DISP=SHR
//PAC70G DD SUBSYS=(&LSR,'DDNAME=SGLSR','BUFND=10')
//SNLSR DD DSN=&INDSV..BVPSN,DISP=SHR
//PAC70N DD SUBSYS=(&LSR,'DDNAME=SNLSR','BUFND=10')
//PAC70S DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),
//          DCB=(RECFM=FB,LRECL=576,BLKSIZE=27648)
//PAC70R DD DSN=&INDSV..BVPSR,DISP=SHR
//SSLR DD DSN=&INDSV..BVPSS,DISP=SHR
//PAC70S DD SUBSYS=(&LSR,'DDNAME=SSLR','BUFND=10')
//PAC70A DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=25200
//PAC70C DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC701 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC702 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC703 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC704 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27180
//PAC706 DD UNIT=&UWK,SPACE=&SPAMAN,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC707 DD UNIT=&UWK,SPACE=&SPAWK,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC708 DD UNIT=&UWK,SPACE=&SPAMAN,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC709 DD UNIT=&UWK,SPACE=&SPAIDX,
//          DCB=(RECFM=FB,LRECL=55,BLKSIZE=12595)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAFLSR DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//SYSPAF DD SUBSYS=(&LSR,'DDNAME=PAFLSR')
//SYSUDUMP DD SYSOUT=&OUT
//PAFP10 EXEC PGM=BVPAPF10,COND=(8,LE,PAC)

```

```

//*------
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//AELSR DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAF80 DD DSN=*.PAC.PAC70B,
// VOL=REF=*.PAC.PAC70B,
// DISP=(OLD,DELETE,DELETE)
//COB80 DD SYSOUT=(&OUT,INTRDR)
//PAFREP DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DEL EXEC PGM=IEFBR14,COND=(8,GT,PAC)
//*------
//PAC7GN DD DSN=*.PAC.PAC70B,
// VOL=REF=*.PAC.PAC70B,
// DISP=(OLD,DELETE,DELETE)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*------
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*------
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)

```

EMLD - Loading of User-Defined Error Messages

EMLD - Introduction

The EMLD procedure performs the initial loading of user- defined error messages. These messages are obtained from the sequential output file of the GPRT procedure (GL-suffixed file).

Execution conditions

Prior execution of GPRT, with an error messages generation request.

EMLD - User Input

One '*' line with user code and password.

EMLD - Description of Steps

Input recognition: PTU001

File definition: IDCAMS

| Code | Physical name | Type | Label |
|--------|------------------|-------|--------------------------|
| PAC7EM | &INDUV..&BASE.EM | Input | User error messages file |

Loading of user-defined error messages in an indexed file: PACL93

| Code | Physical name | Type | Label |
|--------|------------------|--------|--|
| PAC7MB | &&EMLDMB | Input | Input Transactions |
| PAC7GL | &GL | Input | Sequential user-defined error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAC7EM | &INDUV..&BASE.EM | Output | User-defined indexed error messages file |
| PAC7IY | | Report | Output reports |
| PAC7DD | | Report | Authorization control |

Return code :

- 8 : no authorization on batch procedure

EMLD - Execution JCL

```
/** -----  
/**      VISUALAGE PACBASE  
/**  
/** -----  
/**      - LOADING OF USER'S ERROR MESSAGES -  
/**  
/** -----  
/**  
/** THE EMLD PROCEDURE PERFORMS THE INITIAL LOADING OF USER  
/** DEFINED ERROR MESSAGES. THESE MESSAGES ARE OBTAINED  
/** FROM THE SEQUENTIAL OUTPUT FILE OF THE GPRT PROCEDURE  
/** (FILE WITH THE GL SUFFIX).  
/**  
/**  
/** INPUT :
```

```

/** - USER IDENTIFICATION LINE (REQUIRED)
/** -----
/**
//BVPEMLD  PROC BASE=$BASE,           CODE OF DEVPT DATABASE
//          INDSV='$INDSV',           INDEX OF VSAM SYSTEM FILES
//          INDUN='$INDUN',           INDEX OF NON-VSAM USER FILES
//          INDUV='$INDUV',           INDEX OF VSAM USER FILES
//          GL='&&GPRTGL', USER ERROR MESSAGES SEQUENTIAL FILE DSN
/**:       VSAMCAT='$VCAT',           USER VSAM CATALOG
/**:       SYSTCAT='$SCAT',           SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          LSR='BLSR',               LSR BATCH SYSTEM NAME
//          UWK=SYSDA,                WORK UNIT
//          OUT=$OUT                  OUTPUT CLASS
/**-----
//INPUT EXEC PGM=BVPTU001
/**-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&EMLDMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//DEFINE EXEC PGM=IDCAMS
/**-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DF&BASE.EM),DISP=SHR
//PACL93 EXEC PGM=BVPACL93
/**-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**:       DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7EM DD DSN=&INDUV..&BASE.EM,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GL DD DSN=&GL,DISP=SHR
//PAC7MB DD DSN=&&EMLDMB,DISP=(OLD,DELETE)
//PAC7IY DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

EMUP - Update of User-Defined Error Messages

EMUP - Introduction

The EMUP procedure updates the User-Defined Error Message file.

These messages are obtained from the sequential file output by the GPRT procedure (GL-suffixed file) or from transactions for error message deletions at the entity level.

Execution conditions

The User-Defined Error Message file must exist.

In case of the creation and/or modification of error messages, the GPRT procedure must have been executed with the request for the generation of error messages.

EMUP - User Input

A '*' line per library containing entities whose error message(s) must be deleted:

| Position | Length | Value | Meaning |
|----------|--------|----------|---------------|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |
| 19 | 3 | bbb | Library code |

One command line per entity for which error message deletion is requested:

| Position | Length | Value | Meaning |
|----------|--------|-------|--------------------------------------|
| 1 | 1 | 'D' | Transaction code (deletion) |
| 2 | 2 | | Entity type; same as in CHOICE field |
| | | 'O ' | Screen |
| | | 'D ' | Data structure |
| | | 'S ' | Segment |
| 4 | 6 | | Entity code |

EMUP - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Update of indexed user-defined error messages: PACL92

| Code | Physical name | Type | Label |
|--------|-----------------|--------|---|
| PAC7GL | &GL | Input | Sequential user-defined error messages |
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database Data file |
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database index |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database users |
| PAC7MB | &&EMUPMB | Input | Input transactions |
| PAC7EM | &INDUV.&BASE.EM | Output | User-defined error message indexed file |
| PAC7IU | | Report | Transaction report |
| PAC7IX | | Report | Error message report |
| PAC7DD | | Report | Authorization option |

Return code :

- 8 : no batch procedure authorization option.

EMUP - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**          - USER'S ERROR MESSAGES UPDATING -
/**
/** -----
/**
/** THE EMUP PROCEDURE UPDATES THE USER-DEFINED ERROR
/** MESSAGE FILE. THESE MESSAGES ARE OBTAINED FROM THE
/** SEQUENTIAL OUTPUT FILE OF THE GPRT PROCEDURE (FILE WITH
/** A GL SUFFIX) OR FROM TRANSACTIONS FOR ERROR
/** MESSAGE DELETIONS AT THE ENTITY LEVEL.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/** - COMMAND LINE :
/** COL 1  : "D"   TRANSACTION CODE (DELETION)
/** COL 2  : ENTITY TYPE; SAME AS IN CHOICE FIELD.
/**          "O " SCREEN
/**          "D " DATA STRUCTURE
/**          "S " SEGMENT
/** COL 4  : (6 CAR.) ENTITY CODE
/** -----
/**
/**/BVP EMUP  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**          INDSV='$INDSV',                  INDEX OF VSAM SYSTEM FILES

```

```

//      INDSN='$INDSN',          INDEX OF NON-VSAM SYSTEM FILES
//      INDUV='$INDUV',          INDEX OF VSAM USER FILES
//      GL='&&GPRTGL', USER ERROR MESSAGES SEQUENTIAL FILE DSN
//*:    VSAMCAT='$VCAT',          USER VSAM CATALOG
//*:    SYSTCAT='$SCAT',          SYSTEM VSAM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//      LSR='BLSR',              LSR BATCH SYSTEM NAME
//      UWK=$UWK,                WORK UNIT
//      OUT=$OUT                  OUTPUT CLASS
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//CARTE  DD DDNAME=SYSIN
//PAC7MB DD DSN=&&EMUPMB,DISP=(,PASS),
//      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7EM DD DSN=&INDUV..&BASE.EM,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFEM),DISP=SHR
//PACL92 EXEC PGM=BVPACL92
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7EM DD DSN=&INDUV..&BASE.EM,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GL DD DSN=&GL,DISP=SHR
//PAC7IU DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7IX DD SYSOUT=&OUT
//PAC7MB DD DSN=&&EMUPMB,DISP=(OLD,DELETE)
//SYSUDUMP DD SYSOUT=&OUT

```

PPAF - Generated Programs PAF Preprocessor

PPAF - Introduction

Using PAF operators, the PPAF procedure processes generated user programs containing SQL requests for access to the Database.

Execution conditions

None.

Implementation

This procedure may be executed in different ways:

- Either after the generation of programs via GPRT; its generated output is used as input to PPAF, before being compiled or stored in a source program library,
- Or by requesting the procedure in the command lines Before/After generated program; the appropriate JCL must have been previously entered in the selected options (PC screen).

PPAF - User Input

The input is the COBOL source code of programs containing PAF operators to be processed by the pre-processor before being compiled.

After the IDENTIFICATION DIVISION, each program contains a command line for the pre-processor. Its structure is as follows :

| Position | Length | Value | Meaning |
|----------|--------|----------|----------------------------------|
| 1 | 6 | nnnnnn | COBOL line number |
| 7 | 1 | '*' | Comment |
| 8 | 5 | 'TP ' | On-line program OR |
| | | 'BATCH' | Batch program |
| 14 | 5 | 'LIB:' | Fixed label |
| 19 | 3 | bbb | Library code |
| 22 | 1 | blank | Not used |
| 23 | 5 | nnnns | Session number - Session version |
| 28 | 1 | blank | Not used |
| 29 | 2 | -- | Generation variant(s) |
| 32 | 4 | 'AR:' | Fixed label |
| 36 | 1 | 1 | Database language code |
| 38 | 4 | 'SC:' | Batch Language program skeleton |
| | | 'SG:' | On-line program skeleton |
| | | 'SR:' | COBOL program skeleton |
| 42 | 1 | 1 | Skeleton language |
| 43 | 1 | blank | Not used |
| 44 | 6 | 'SINGLE' | Single quotes OR |

| Position | Length | Value | Meaning |
|----------|--------|----------|---------------|
| | | 'DOUBLE' | Double quotes |

Examples

000020*TP LIB: APP 2345 00 AR: F SG: F SINGLE

000020*BATCH LIB: APP 2300T 4 AR: F SC: F DOUBLE

This line is automatically generated by the GPRT procedure.

Printed output

This procedure prints an error report.

Result

The result of the PPAF procedure is the COBOL source in which PAF operators have been processed and calls to PAF batch or on-line sub-programs have been generated.

PPAF - Description of Steps

Check of VSAM files: IDCAMS

Preprocessor: PAFP10

| Code | Physical name | Type | Label |
|--------|------------------|--------|-----------------------------------|
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAF80 | &PAFINPUT | Input | Generated programs |
| COB80 | &&PAFGEN | Output | Generated programs to be compiled |
| PAFREP | | Report | Error report |

PPAF - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - VA PAC ACCESS FACILITY PRE-PROCESSING -
/**
/** -----
/**

```

```

/** USING PAF OPERATORS, THE PPAF PROCEDURE PROCESSES
/** GENERATED USER PROGRAMS CONTAINING SQL REQUESTS FOR
/** ACCESS TO THE DATABASE.
/**
/** USER INPUT IS THE COBOL SOURCE CODE OF PROGRAMS
/** CONTAINING PAF OPERATORS TO BE PROCESSED BY
/** BY THE PRE-PROCESSOR BEFORE COMPILATION.

/**
/** -----
/**
/**BVPPPAF  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
/**          INDSN='$INDSN',                INDEX OF SYSTEM NON VSAM FILES
/**          INDUV='$INDUV',                INDEX OF USER VSAM FILES
/***:       VSAMCAT='$VCAT',                USER VSAM CATALOG
/***:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
/**          STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
/**          OUT=$OUT,                       OUTPUT CLASS
/**          PAFINPUT=,                     DSNNAME OF PAF INPUT TABLES
/**          SPAMB='(TRK,(150,15))',        SPACE OF GENERATED PROGRAMS
/**          UWK=$UWK                        WORK UNIT
/*******
/**VERIFY EXEC PGM=IDCAMS
/**-----
/***:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/***:       DD DSN=&SYSTCAT,DISP=SHR
/**SYSPRINT DD SYSOUT=&OUT
/**PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
/**PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
/**SYSIN    DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
/**         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
/**PAFP10   EXEC PGM=BVPAFP10
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**         DD DSN=$BCOB,DISP=SHR
/***:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/***:       DD DSN=&SYSTCAT,DISP=SHR
/**PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
/**PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
/**PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
/**PAF80    DD DSN=&PAFINPUT,DISP=SHR
/**COB80    DD DSN=&&PAFGEN,DISP=(,PASS),
/**         UNIT=&UWK,SPACE=&SPAMB,
/**         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**PAFREP   DD SYSOUT=&OUT
/**SYSOUT   DD SYSOUT=&OUT
/**SYSUDUMP DD SYSOUT=&OUT

```

GPRC - COBOL API management

GPRC - Introduction

GPRC and the COBOL API: GPRC

This procedure makes it possible to use Client/Server services, such as Folders and Elementary components, in batch mode.

To do this, the GPRT procedure is completed by specific processing whose result is the GPRC procedure. It consists in the generation of sources for the COBOL API of the Folder manager. GPRC is a procedure dedicated to this type of generation ONLY.

For more information refer to the 'COBOL API User's Guide' manual.

GPRC - User Input

Refer to the description of GPRT user input.

GPRC - Description of Steps

Generation and Print: PACB

The generated source provided depends on the generation-print commands taken into account.

The entities which can use the COBOL API are:

- Programs,
- Macrostructures,
- Screens,
- Elementary Components.

For a complete information, refer to the GPRT description.

COBOL API extractor: PAPG1S

| Code | Physical name | Type | Label |
|----------|------------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| GENERE | *.PAC.PAC7OB | Input | GPRT generated source |
| PAC7W1 | &&PAC7W1 | Output | Work file |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

COBOL API: PAPG5S

| Code | Physical name | Type | Label |
|----------|---------------|--------|-------------|
| PAC7W1 | &&PAC7W1 | Input | Input file |
| PAC7W2 | &&PAC7W2 | Output | Output file |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

COBOL API generator: PAPG7S

| Code | Physical name | Type | Label |
|----------|----------------|--------|------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PAC7EW | &&PAC7EW | Output | Generation errors file |
| PAC7W2 | &&PAC7W2 | Input | Intermediate file |
| PAC7W3 | &&PAC7W3 | Input | Intermediate file |
| PAC7SA | &INDSV..BVP SA | Input | Skeleton of COBOL API labels |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

COBOL API - COBOL insertion: PAPG9S

| Code | Physical name | Type | Label |
|--------|----------------|--------|----------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PAC7EW | &&PAC7EW | Input | Generation errors file |
| GENERE | *.PAC.PAC7OB | Input | GPRT generated source |
| PAC7W3 | &&PAC7W3 | Output | Intermediate file |
| COB80 | &&COB80 | Input | Generated API COBOL source |
| PAC7ED | | Report | Errors report |

GPRC - Execution JCL

```
/** -----  
/**      VISUALAGE PACBASE  
/**  
/** -----  
/**      - GENERATION AND PRINTING WITH API COBOL -  
/**  
/** -----
```

```

/**
/** IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST
/** CONTAIN THE JCL REQUIRED TO COMPILE THEM,
/** USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND
/** THE BEFORE/AFTER PROGRAM OPTIONS.
/**
/** -----
/**
//BVPGPRC PROC BASE=$BASE, CODE OF VAPAC DATABASE
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES
/**: VSAMCAT='$VCAT', USER VSAM CATALOG
/**: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// COPIES=1, NUMBER OF REPORT COPIES
// NBMAN=1, NUMBER OF COPIES OF VOLUME (PDM)
// OUT=$OUT, UTILITIES AND ERRORS OUTPUT CLASS
// OUTL=$OUT, OUTPUT CLASS OF REPORTS
// VOL='SER=$VOLUN', VOLUME OF GENERATION FILE
// UNIT=$UNITUN, DISK UNIT OF GENERATION FILE
// UWK=$UWK, WORK UNIT
// USER=, USER CODE
// SCMMVT='NO', SCM TRANSACTION
// APPLID='', CICS APPLID
// LSR='BLSR', LSR BATCH SYSTEM NAME
// SPAGN='(TRK,(100,10))', GENERATION FILE SPACE
// SPAEX='(TRK,(50,10),RLSE)', SPACE OF PRINTING FILES
// SPAWK='(TRK,(50,20))', WORK FILE SPACE
// SPAMB='(TRK,(5,1),RLSE)', REQUEST FILE SPACE
// SPAMAN='(TRK,(50,10),RLSE)', VOLUME (PDM) FILE SPACE
// SPAWIN='(TRK,(50,10),RLSE)', WINDOWING FILE SPACE
// SPAIDX='(TRK,(50,10),RLSE)' INDEX OF VOLUME FILE SPACE
/**-----
//INPUT EXEC PGM=BVPTU001
/**-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&GPRTMB,DISP=(,PASS),UNIT=&UWK,
// DCB=BLKSIZE=3440,SPACE=&SPAMB
//CARTE DD DDNAME=SYSIN
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
/**-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYSYPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
/**-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR

```

```

//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYPAPF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR

//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PAC7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFLB),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFQJ),DISP=SHR
//PAC EXEC PGM=BVPACB,REGION=0K,
// PARM='&SCMMVT,&BASE,&APPLID,BVP'
//*-----
/*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
/*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//* DD DSN=$DFHEXC,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//AELSR DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PAC7AJ DD DUMMY
/*AJLSR DD DSN=&INDUV..&BASE.AJ,DISP=SHR
/*PAC7AJ DD SUBSYS=(&LSR,'DDNAME=AJLSR')
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR')
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PACGGK DD DSN=&INDSV..BVPGK,DISP=SHR

```

```

//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR')
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7DG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=12560
//PAC7EB DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EE DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EI DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7EP DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EQ DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7ER DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EW DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7GB DD DSN=&&PAC7GB,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GD DD DSN=&&PAC7GD,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GE DD DSN=&&PAC7GE,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GF DD DSN=&&PAC7GF,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GG DD DSN=&&PAC7GG,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GI DD DSN=&&PAC7GI,UNIT=&UWK,DCB=BLKSIZE=3440,
// SPACE=(TRK,(10,5),RLSE),DISP=(,DELETE)
//PAC7GK DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GL DD DSN=&&PAC7GL,UNIT=&UWK,
// SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=1800
//PAC7GM DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GN DD DSN=&&PAC7GN,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=265,BLKSIZE=6095),
// SPACE=&SPAMAN
//PAC7GO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=4140
//PAC7GP DD DSN=&&PAC7GP,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GQ DD DSN=&&PAC7GQ,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GR DD DSN=&&PAC7GR,UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
// SPACE=&SPAGN,DISP=(,PASS)
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//PAC7GT DD DSN=&&PAC7GT,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=260,BLKSIZE=7800),
// SPACE=&SPAWIN
//PAC7GV DD DSN=&&PAC7GV,UNIT=&UWK,

```

```

//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//          SPACE=&SPAGN,DISP=(,PASS)
//PAC7G6 DD DSN=&&PAC7G6,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=368,BLKSIZE=27232),
//          SPACE=&SPAMAN
//PAC7DB DD SYSOUT=&OUTL
//PAC7IA DD SYSOUT=&OUTL
//PAC7ID DD SYSOUT=&OUTL,COPIES=&COPIES
//PAC7IK DD SYSOUT=&OUTL
//PAC7IL DD SYSOUT=&OUTL
//PAC7IN DD SYSOUT=&OUTL,COPIES=&NBMAN
//PAC7IO DD SYSOUT=&OUTL
//PAC7IW DD SYSOUT=&OUTL
//PAC7JG DD DSN=&&PAC7JG,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440),
//          SPACE=&SPAMB
//PAC7KB DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KD DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KE DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KF DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KM DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7KP DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KQ DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KR DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KS DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KU DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7LG DD DUMMY,DCB=BLKSIZE=90
//PAC7LI DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3440
//PAC7LK DD DUMMY,DCB=BLKSIZE=100
//PAC7LM DD DUMMY,DCB=BLKSIZE=100
//PAC7ME DD DSN=&&GPRTMB,DISP=(OLD,DELETE,DELETE)
//PAC7MG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3060
//PAC7MV DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=6800
//PAC7OB DD DSN=&INDUN..&BASE..GN&USER,
//          DCB=BLKSIZE=12560,
//          UNIT=&UNIT,
//          VOL=&VOL,
//          SPACE=&SPAGN,DISP=(,PASS,DELETE)
//PAC7OD DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OE DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OF DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OG DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,
//          DISP=(MOD,,DELETE)
//PAC7OP DD DSN=*.PAC7OB,
//          VOL=REF=*.PAC7OB,

```

```

//          DISP=(MOD,,DELETE)
//PAC70Q DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70R DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//PAC70V DD DSN=*.PAC70B,
//          VOL=REF=*.PAC70B,
//          DISP=(MOD,,DELETE)
//QJLSR DD DSN=&INDSV..BVPQJ,DISP=SHR
//PAC7QJ DD SUBSYS=(&LSR,'DDNAME=QJLSR','BUFND=10')
//PAC7SC DD DSN=&INDSV..BVPSC,DISP=SHR
//SGLSR DD DSN=&INDSV..BVPSG,DISP=SHR
//PAC7SG DD SUBSYS=(&LSR,'DDNAME=SGLSR','BUFND=10')
//SNLSR DD DSN=&INDSV..BVPSN,DISP=SHR
//PAC7SN DD SUBSYS=(&LSR,'DDNAME=SNLSR','BUFND=10')
//PAC7SO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),
//          DCB=(RECFM=FB,LRECL=576,BLKSIZE=27648)
//PAC7SR DD DSN=&INDSV..BVPSR,DISP=SHR
//SSLR DD DSN=&INDSV..BVPSS,DISP=SHR
//PAC7SS DD SUBSYS=(&LSR,'DDNAME=SSLR','BUFND=10')
//PAC7WA DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=25200
//PAC7WC DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W1 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W2 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W3 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W4 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27180
//PAC7W6 DD UNIT=&UWK,SPACE=&SPAMAN,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W7 DD UNIT=&UWK,SPACE=&SPAWK,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W8 DD UNIT=&UWK,SPACE=&SPAMAN,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W9 DD UNIT=&UWK,SPACE=&SPAIDX,
//          DCB=(RECFM=FB,LRECL=55,BLKSIZE=12595)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAFLSR DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//SYSPAF DD SUBSYS=(&LSR,'DDNAME=PAFLSR')
//SYSUDUMP DD SYSOUT=&OUT
//PAPG1S EXEC PGM=BVPAPG1S,COND=(8,LE,PAC)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//AELSR DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR

```

```

//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//GENERE DD DSN=*.PAC.PAC70B,
//        DISP=(OLD,PASS,DELETE)
//PAC7W1 DD DSN=&&PAC7W1,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=260,BLKSIZE=2600),
//        SPACE=&SPAWK
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAPG5S EXEC PGM=BVPAPG5S,COND=(8,LE,PAC)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PAC7W1 DD DSN=&&PAC7W1,DISP=(OLD,DELETE)
//PAC7W2 DD DSN=&&PAC7W2,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=260,BLKSIZE=2600),
//        SPACE=&SPAWK
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAPG7S EXEC PGM=BVPAPG7S,COND=(8,LE,PAC)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//AELSR DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//SALSR DD DSN=&INDSV..BVPSA,DISP=SHR
//PAC7SA DD SUBSYS=(&LSR,'DDNAME=SALSR','BUFND=10','BUFNI=10')
//PAC7W2 DD DSN=&&PAC7W2,DISP=(OLD,DELETE)
//PAC7EW DD DSN=&&PAC7EW,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=260,BLKSIZE=2600),
//        SPACE=&SPAWK
//PAC7W3 DD DSN=&&PAC7W3,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//        SPACE=&SPAWK
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAPG9S EXEC PGM=BVPAPG9S,COND=(8,LE,PAC)

```



```

//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//AELSR DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7AE DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PAC7EW DD DSN=&&PAC7EW,DISP=(OLD,DELETE)
//PAC7W3 DD DSN=&&PAC7W3,DISP=(OLD,DELETE)
//GENERE DD DSN=*.PAC.PAC70B,
// DISP=(OLD,PASS,DELETE)
//COB80 DD DSN=&&COB80,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
// SPACE=&SPAWK
//PAC7ED DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)

```

GPMC - Management of MCI Operator

GPMC - Introduction

Generation and Move Corresponding

The MCI operator entered in VA Pacbase is not interpreted by the PACB generator, but later by the two programs (UTIMCR and UTIMCI) which process the COBOL code output by the generator.

Basic rules:

The MOVE CORRESPONDING applies to two group fields: the first operand must be entered on the same line as the operator, and the second operand must be entered next to the first one or on a continuation line.

Each elementary element of the first group must have its equivalent in the second group to be included in the MOVE.

The comparison of the COBOL fields is based on their 'Data Element' code, i.e. the character string after the first dash (there is a prefix before this dash).

For example, in 'PREFIX-FIELD-NUMBER-ONE', the whole string 'FIELD-NUMBER-ONE' is searched for in the composition of the other group ; if it is found, it will be included in the MOVE CORRESPONDING.

When a group is followed by an index, the fields which are generated are generated in the same way.

Since none of the 'MOVE' statements generated in this way is controlled, the errors, if any, will be detected by the COBOL compiler.

The WKMCI file, written by UTIMCR, lists the lines of the MCI statements detected in the analyzed COBOL (one or two lines for each statement, depending on the user input) and is read by the UTIMCI program. The COB80 file which contains the generated COBOL is read by the two programs. The final file (MCI80) is the image of the COB80 file ; in the original COBOL, it copies the MCI lines as comments followed by the induced MOVE statements.

GPMC - User Input

Refer to the description of GPRT user input.

GPMC - Description of Steps

Generation and Print: PACB

The provided generated source depends on the generation-print commands taken into account.

For more information, refer to the GPRT description.

MCI generator: UTIMCR

| Code | Physical name | Type | Label |
|-------|---------------|--------|---------------------|
| MCI80 | *.PAC.PAC7OB | Input | GPRT generated code |
| WKMCI | &&GPMCWK | Output | Work file |

Return codes:

- 4 : No MCI statement has been detected and the processing stops.

MCI generator: UTIMCI

| Code | Physical name | Type | Label |
|----------|---------------|--------|---------------------|
| MCI80 | *.PAC.PAC7OB | Input | GPRT generated code |
| WKMCI | &&GPMCWK | Input | Work file |
| COB80 | &&COB80 | Output | Result source |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return code:

- 8: Unknown Source or Target of an MCI operator

GPMC - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - MOVE CORRESPONDING GENERATION -
/**
/** -----
/** IN ADDITION TO THE GENERATED ENTITIES, THE FILE MUST
/** CONTAIN THE JCL REQUIRED TO COMPILE THEM,
/** USING THE BEGINNING/END OF JCL JOB STREAM OPTIONS AND
/** THE BEFORE/AFTER PROGRAM OPTIONS.
/**
/** -----
/**
/**BVPGPMC  PROC BASE=$BASE,                CODE OF VAPAC DATABASE
/**          INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES
/**          INDSN='$INDSN',                INDEX OF NON-VSAM SYSTEM FILES
/**          INDUV='$INDUV',                INDEX OF VSAM USER FILES
/**          INDUN='$INDUN',                INDEX OF USER NON VSAM FILES
/***:       VSAMCAT='$VCAT',                USER VSAM CATALOG
/***:       SYSCAT='$SCAT',                SYSTEM VSAM CATALOG
/**          STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
/**          SORTLIB='$BIBT',              SORT LIBRARY
/**          COPIES=1,                      NUMBER OF REPORT COPIES
/**          NBMAN=1,                       NUMBER OF COPIES OF VOLUME (PDM)
/**          OUT=$OUT,                       UTILITIES AND ERRORS OUTPUT CLASS
/**          OUTL=$OUT,                      OUTPUT CLASS OF REPORTS
/**          VOL='SER=$VOLUN',              VOLUME OF GENERATION FILE
/**          UNIT=$UNITUN,                  DISK UNIT OF GENERATION FILE
/**          UWK=$UWK,                      WORK UNIT
/**          USER=,                          USER CODE
/**          SCMMVT='NO',                   SCM TRANSACTION
/**          APPLID='',                     CICS APPLID
/**          LSR='BLSR',                    LSR BATCH SYSTEM NAME
/**          SPAGN='(TRK,(100,10))',       GENERATION FILE SPACE
/**          SPAEX='(TRK,(50,10),RLSE)',   SPACE OF PRINTING FILES

```

```

//          SPAWK='(TRK,(50,20))',           WORK FILE SPACE
//          SPAMB='(TRK,(5,1),RLSE)',        REQUEST FILE SPACE
//          SPAMAN='(TRK,(50,10),RLSE)',     VOLUME (PDM) FILE SPACE
//          SPAWIN='(TRK,(50,10),RLSE)',     WINDOWING FILE SPACE
//          SPAIDX='(TRK,(50,10),RLSE)' INDEX OF VOLUME FILE SPACE
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&GPRTMB,DISP=(,PASS),UNIT=&UWK,
//          DCB=BLKSIZE=3440,SPACE=&SPAMB
//CARTE DD DDNAME=SYSIN
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYSYPAF,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSYPAF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSYPAF DD DSN=&INDUV..SYSYPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//PAC7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFLB),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFQJ),DISP=SHR

```

```

//PAC      EXEC PGM=BVPACB,REGION=0K,
//          PARM='&SCMMVT,&BASE,&APPLID,BVP'
//*-----
//*:STPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//*      DD DSN=$DFHEXC,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSOUX  DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//AELSR   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE  DD SUBSYS=(&LSR,'DDNAME=AELSR')
//PAC7AJ  DD DUMMY
//*AJLSR  DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//*PAC7AJ DD SUBSYS=(&LSR,'DDNAME=AJLSR')
//ANLSR   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN  DD SUBSYS=(&LSR,'DDNAME=ANLSR')
//ARLSR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR  DD SUBSYS=(&LSR,'DDNAME=ARLSR')
//AYLSR   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY  DD SUBSYS=(&LSR,'DDNAME=AYLSR')
//PAC7LB  DD DSN=&INDSV..BVPLB,DISP=SHR
//PACGGK  DD DSN=&INDSV..BVP GK,DISP=SHR
//GNLSR   DD DSN=&INDSV..BVP GN,DISP=SHR
//PACGGN  DD SUBSYS=(&LSR,'DDNAME=GNLSR')
//GRLSR   DD DSN=&INDSV..BVP GR,DISP=SHR
//PACGGR  DD SUBSYS=(&LSR,'DDNAME=GRLSR')
//PACGGU  DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7DG  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=12560
//PAC7EB  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EE  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EG  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EI  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=6300
//PAC7EN  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7EP  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EQ  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7ER  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EV  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EW  DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7GB  DD DSN=&&PAC7GB,UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//          SPACE=&SPAGN,DISP=(,PASS)
//PAC7GD  DD DSN=&&PAC7GD,UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//          SPACE=&SPAGN,DISP=(,PASS)
//PAC7GE  DD DSN=&&PAC7GE,UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//          SPACE=&SPAGN,DISP=(,PASS)
//PAC7GF  DD DSN=&&PAC7GF,UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//          SPACE=&SPAGN,DISP=(,PASS)
//PAC7GG  DD DSN=&&PAC7GG,UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//          SPACE=&SPAGN,DISP=(,PASS)

```

```

//PAC7GI DD DSN=&&PAC7GI,UNIT=&UWK,DCB=BLKSIZE=3440,
//        SPACE=(TRK,(10,5),RLSE),DISP=(,DELETE)
//PAC7GK DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GL DD DSN=&&PAC7GL,UNIT=&UWK,
//        SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=1800
//PAC7GM DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=2000
//PAC7GN DD DSN=&&PAC7GN,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=265,BLKSIZE=6095),
//        SPACE=&SPAMAN
//PAC7GO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=4140
//PAC7GP DD DSN=&&PAC7GP,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7GQ DD DSN=&&PAC7GQ,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7GR DD DSN=&&PAC7GR,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//PAC7GT DD DSN=&&PAC7GT,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=260,BLKSIZE=7800),
//        SPACE=&SPAWIN
//PAC7GV DD DSN=&&PAC7GV,UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=12560),
//        SPACE=&SPAGN,DISP=(,PASS)
//PAC7G6 DD DSN=&&PAC7G6,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=368,BLKSIZE=27232),
//        SPACE=&SPAMAN
//PAC7DB DD SYSOUT=&OUTL
//PAC7IA DD SYSOUT=&OUTL
//PAC7ID DD SYSOUT=&OUTL,COPIES=&COPIES
//PAC7IK DD SYSOUT=&OUTL
//PAC7IL DD SYSOUT=&OUTL
//PAC7IN DD SYSOUT=&OUTL,COPIES=&NBMAN
//PAC7IO DD SYSOUT=&OUTL
//PAC7IW DD SYSOUT=&OUTL
//PAC7JG DD DSN=&&PAC7JG,DISP=(,PASS),UNIT=&UWK,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440),
//        SPACE=&SPAMB
//PAC7KB DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KD DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KE DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KF DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KM DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KN DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=9100
//PAC7KP DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KQ DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KR DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KS DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KU DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7LG DD DUMMY,DCB=BLKSIZE=90
//PAC7LI DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3440

```

```

//PAC7LK DD DUMMY,DCB=BLKSIZE=100
//PAC7LM DD DUMMY,DCB=BLKSIZE=100
//PAC7ME DD DSN=&&GPRTMB,DISP=(OLD,DELETE,DELETE)
//PAC7MG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3060
//PAC7MV DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=6800
//PAC7OB DD DSN=&INDUN..&BASE..GN&USER,
// DCB=BLKSIZE=12560,
// UNIT=&UNIT,
// VOL=&VOL,
// SPACE=&SPAGN,DISP=(,PASS,DELETE)
//PAC7OD DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OE DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OF DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OG DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OP DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OQ DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OR DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//PAC7OV DD DSN=*.PAC7OB,
// VOL=REF=*.PAC7OB,
// DISP=(MOD,,DELETE)
//QJLSR DD DSN=&INDSV..BVPQJ,DISP=SHR
//PAC7QJ DD SUBSYS=(&LSR,'DDNAME=QJLSR','BUFND=10')
//PAC7SC DD DSN=&INDSV..BVPSJ,DISP=SHR
//SGLSR DD DSN=&INDSV..BVPSG,DISP=SHR
//PAC7SG DD SUBSYS=(&LSR,'DDNAME=SGLSR','BUFND=10')
//SNLSR DD DSN=&INDSV..BVPSN,DISP=SHR
//PAC7SN DD SUBSYS=(&LSR,'DDNAME=SNLSR','BUFND=10')
//PAC7SO DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),
// DCB=(RECFM=FB,LRECL=576,BLKSIZE=27648)
//PAC7SR DD DSN=&INDSV..BVPSR,DISP=SHR
//SSLR DD DSN=&INDSV..BVPSS,DISP=SHR
//PAC7SS DD SUBSYS=(&LSR,'DDNAME=SSLR','BUFND=10')
//PAC7WA DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=25200
//PAC7WC DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W1 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W2 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W3 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W4 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27180
//PAC7W6 DD UNIT=&UWK,SPACE=&SPAMAN,
// DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W7 DD UNIT=&UWK,SPACE=&SPAWK,

```

```

//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W8 DD UNIT=&UWK,SPACE=&SPAMAN,
//          DCB=(RECFM=FB,LRECL=331,BLKSIZE=29128)
//PAC7W9 DD UNIT=&UWK,SPACE=&SPAIDX,
//          DCB=(RECFM=FB,LRECL=55,BLKSIZE=12595)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAFLSR DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//SYSPAF DD SUBSYS=(&LSR,'DDNAME=PAFLSR')
//SYSUDUMP DD SYSOUT=&OUT
//UTIMCR EXEC PGM=BVPUTMCR
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//MCI80 DD DSN=*.PAC.PAC70B,
//          DISP=(OLD,PASS)
//WKMCI DD DSN=&&GPMCWK,DISP=(,PASS),
//          UNIT=&UNIT,VOL=&VOL,
//          SPACE=&SPAWK,
//          DCB=(RECFM=FB,BLKSIZE=3281,LRECL=193)
//UTIMCI EXEC PGM=BVPUTMCI,COND=(0,NE,UTIMCR)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//MCI80 DD DSN=*.PAC.PAC70B,
//          DISP=(OLD,PASS,DELETE)
//WKMCI DD DSN=&&GPMCWK,DISP=(OLD,DELETE)
//COB80 DD DSN=&&COB80,DISP=(,CATLG),
//          UNIT=&UNIT,VOL=&VOL,
//          SPACE=&SPAWK,
//          DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR

```



```
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
/*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYS PRINT DD SYSOUT=&OUT
//SYS IN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)
```

Chapter 3. Extractions

PACX - Introduction

The extraction procedure allows to perform various types of data extractions from the Development Database via a PAF extractor (selection of criteria).

See chapter 'UPDP - Update from PAF Tables' in 'The Developer's Procedures' manual.

Data is extracted as transactions that can be used as input to the following procedures:

- UPDT
- UPDP
- CPSN (If the optional 'Partitioned Database Manager' utility is available.)

Execution conditions

None since the Database is not directly updated by this procedure.

PACX - User Input Common to all Extractors

| Position | Length | Value | Meaning |
|----------|--------|----------|---|
| 2 | 1 | '* | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | Password |
| 19 | 3 | bbb | Extraction library code, or target Library code if RMEN with upload |
| 22 | 4 | nnnn | Session number (blank=current ses.) |
| 26 | 1 | 'T' | Session status if Test session |
| 29 | 4 | cccc | Extractor code (1) |
| 33 | 1 | '1' | Formatting for UPDT |
| | | '2' | CPSN : formatting for UPDT with explicit transaction codes |
| | | ' ' | No formatting for UPDT |
| 34 | 1 | '1' | Formatting for UPDP (PAF) |
| | | '2' | CPSN : formatting for UPDP with explicit transaction codes |

| Position | Length | Value | Meaning |
|----------|--------|--------|---|
| | | ' ' | No formatting for UPDP (PAF) |
| 35 | 1 | '1' | Formatting for CPSN |
| | | ' ' | No formatting for CPSN |
| 40 | 3 | ppp | DSMS Product Code |
| 43 | 6 | nnnnnn | DSMS Change number (DSMS Function only) |
| 49 | 1 | | Lock processing |
| | | ' ' | Lock extraction with user code ('*-line user code) |
| | | '1' | No lock extraction |
| | | '2' | Lock extraction with user code (original user code) |
| | | 'N' | For RMEN only : no extraction of locked entities by an other user |
| 50 | 1 | ' ' | No transfer of password |
| | | '1' | Password transfer |
| 69 | 3 | bbb | Library code for the '*-line of the output file(s) (For EXTR, EXLI and EXUE only) |
| 76 | 5 | nnnnT | Session number for the '*-line of the output file(s) (For EXTR, EXLI and EXUE only) |

(1) The possible values for the extractor code are:

- EXTR: Extraction of entities (extracted transactions are sorted).
- EXTA: Extraction of entities (extracted transactions are sorted, according to the input identification lines order. So if each request is preceded by a '*' line, extracted transactions will be sorted in the order of the requests). The formatting is forced to UPDT.
- EXUE: Extraction of user entities
- EXLI: Extraction of libraries or library sub-networks (formatting for UPDP, UPDT or CPSN).
- EXPJ: Extraction of Journal (formatting for CPSN is not possible)
- EXPU: Extraction for purge (formatting for CPSN is not possible)
- RMEN: Extraction of entities for upload/replacement/ recoding (formatting for CPSN is not possible). RMEN is subject to a separate purchase agreement.
- CPSN: comparison of sub-networks or entities.

Important

- One extractor type only for each run: If the procedure detects more than one type of extractors, it will take only the first one into account.
- The formatting type of the first '*' line only is taken into account.
- Formatting for CPSN: This procedure is part of the 'Partitioned Database Manager' optional utility. Its use is therefore subject to a separate purchase agreement.
- Maximum number of input '*' lines : 1 for RMEN and EXPJ, 1000 for EXTR, EXTA, EXUE and EXPU.
- The Pacdesign/Pacbench technical locks are not extracted by PACX.

Results

The PACX procedure produces:

- A report which contains the list of executed programs and the number of generated transactions.
- A list of requests with possible associated errors.
- One or more execution reports depending on the type of extractor.

EXTR/EXTA - Extraction of Entities

EXTR/EXTA - Introduction

These extractor types allow the selection of all or only part of an entity.

If the request has an 'ALL' type, the whole entity is extracted, i.e. the entity itself but also all the entities it uses, as well as entities used by those, and so on. Used entities that are not cross-referenced are not extracted.

Depending on the type of formatting requested, the resulting file can be used as input to the UPDT, UPDP or CPSN procedures (if the request is of the 'ALL', 'ONLY' or 'EXPT' type; the formatting for CPSN is not allowed).

For EXTA, the formatting is forced to UPDT.

EXTR/EXTA - User Input

One or two command lines per entity to be extracted.

First line :

| Position | Length | Value | Meaning |
|----------|--------|-------|-------------|
| 2 | 1 | 'W' | Line code |
| 3 | 1 | '1' | Line number |
| 4 | 2 | 'EX' | |

| Position | Length | Value | Meaning |
|----------|--------|--------|--|
| 6 | 1 | | Library selection code: |
| | | 'U' | Library alone |
| | | 'C' | Library and its upper-level libraries |
| | | '+' | Library and its upper-level libraries with identification lines ('*' lines) generation |
| 7 | 33 | Choice | Entity to be extracted, coded in the same way as the 'Choice' field in TP. |
| 40 | 4 | | Extraction type (1): |
| | | ' ' | Entity alone (required for EXTA) |
| | | 'ALL ' | Entity and used entities |
| | | 'ONLY' | Entity and only used entities whose types are specified in the following part of the line |
| | | 'EXPT' | Entity and used entities, except those whose types are specified in the following part of the line |
| 44 | | | 12-position table (3 char./position) containing exceptions or selections |
| | | | 'DEL': Element |
| | | | 'DBD': Database Block |
| | | | 'DST': Data Structure |
| | | | 'SEG': Segment |
| | | | 'RPT': Report |
| | | | 'TXT': Text |
| | | | 'VOL': PDM Volume |
| | | | 'PGM': Program |
| | | | 'DLG': Dialog |
| | | | 'SCR': Screen |
| | | | 'PIA': Parameterized Input Aid |
| | | | 'MET': Methodology |
| | | | 'CME': Client Meta-Entity |
| | | | 'CLR': Client User Relation |
| | | | '\$tt': User Entity (tt = Meta-entity type) |
| | | | 'EME': Extension Meta-Entity |
| | | | 'ERL : Extension User Relation |
| | | | 'Ytt': Extension User Entity (tt = Meta-Entity type (1)) |

(1) The use of the 'ALL', 'ONLY' and 'EXPT' values is submitted to the acquisition of the 'In-Depth extractor' optional module.

Second line (continuation line for selections and exceptions):

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 2 | 1 | 'W' | Line code |
| 3 | 1 | '2' | Line number |
| 44 | | | 12-position table (3 characters per position) containing the exceptions or selections |

(1) The Meta-Entity type are :

| CE | Elementary Component |
|----|--------------------------|
| CS | eBusiness Application |
| C1 | SCM tool interface |
| DO | Folder |
| D1 | Publishing Documents |
| D2 | Document Type Definition |
| F1 | External Files |
| G1 | Generation-print command |
| MC | Communication Monitor |
| MS | Message |
| OP | Operation |
| PT | Part |
| SB | SOAP Binding |
| SI | Initialization Server |
| SV | Service |
| UM | UML Interface |
| VL | Logical View |
| 5Q | Quality rule definition |
| 7E | Extraction master path |

The EXTR procedure also works with choices that are specific to the Development Database.

These choices must be entered from the seventh position, in the following way:

//A_CCCXXXXXX

where A is the methodology code and CCC the entity local code.

Type of extraction

- The 'in-depth extractor' option ('ALL', 'EXPT' or 'ONLY' extraction type) is not available for EXTA. For this procedure, the value must be blank.
- By default, the extraction of a Data Structure extracts its Segments. To prevent the Segments from being extracted, you must enter 'EXPTSEG' as the extraction type. This is possible, even if the 'in-depth extractor' option is not available.
- The extraction of a Dialog extracts only the Dialog by default. To extract the Dialog 's screens, enter 'ALL'.
- Same as above for the extraction of a Meta-Entity and its User Entities.
- The extraction stops at the first level of selection or exception.

Example: Extraction of a Program with 'EXPTSEG' - The Elements used by the Segments used by the Program are not extracted since the extractor does not consider those segments.

Printed output

The procedure produces a list of extracted entities:

- Sorted for EXTR,
- In the order of the requests for EXTA.

EXUE - Extraction of User Entities Contents

EXUE - Introduction

The EXUE procedure extracts the contents of User Entities according to the Meta-Entity type code, formatted as simple records in a sequential file.

The EXUE procedure is part of the Dictionary Extensibility Function which is an optional component and whose use is subject to a separate purchase agreement.

See the 'Dictionary Extensibility' Manual.

EXUE - User Input

One command line per user entity:

| Position | Length | Value | Meaning |
|----------|--------|--------|---------------------------------|
| 2 | 4 | 'W1EX' | Line code |
| 6 | 1 | '\$' | Client UE Extraction identifier |

| Position | Length | Value | Meaning |
|----------|--------|-------|---------------------------------------|
| | | 'Y' | Extension UE Extraction identifier |
| 7 | 1 | | Library selection code: |
| | | 'U' | Selected Library |
| | | 'C' | Selected Library + higher level Libr. |
| 8 | 2 | CC | Meta-Entity call type |

Printed output

The EXUE procedure prints a list of the extracted UEs.

Result

The output of the EXUE procedure is a sequential file with a fixed format in which the contents of the selected user entities are recorded.

The length of each record is 230 characters.

Each record includes:

- A common part containing all the characteristics necessary to identify each extracted line.
- A specific part whose format depends on the meta-entity description.

PACX - Description of Steps

Input recognition: PTU001

Initialization of the KSDS work file: IDCAMS

Check of VSAM files: IDCAMS

Extraction: PACX

This step extracts transactions according to user input.

| Code | Physical name | Type | Label |
|--------|------------------|-------|----------------------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database index |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database data |

| Code | Physical name | Type | Label |
|--------|----------------------|--------------|--|
| PAC7AY | &INDUV..&BASE.AY | Input | Development Database extension data |
| PACGGN | &INDSV..BVPGN | Input | Administration Database index |
| PACGGR | &INDSV..BVPGR | Input | Administration Database data |
| PACGGU | &INDSV..BVPGU | Input | Administration Database users |
| PAC7PJ | &INDUN..&BASE.PJ(0) | Input | Archived transactions |
| PAC7MB | &&PACXMB | Input | User input |
| PAC7MA | DUMMY | Input | CPSN Master file |
| PAC7ES | DUMMY | Input | CPSN Slave file |
| PAC7BM | &&PACXBM | Input/Output | User input |
| PAC7MM | &&PACXMM | Input/Output | EXPU Work file |
| PAC7MJ | &&PACXMJ | Input/Output | EXPJ Work file |
| PAC7TE | &&PACXTE | Input/Output | RMEN Work file |
| PAC7RE | &&PACXRE | Input/Output | RMEN Work file |
| PAC7RM | &&PACXRM | Input/Output | RMEN Work file |
| PAC7WD | &&PACXWD | Input/Output | Extracted transactions |
| SYSEXT | &INDUV..SYSEXT.&USER | Input/Output | Work file |
| PAC7RQ | DUMMY | Input/Output | Work file |
| PAC7MV | &&MV | Output | Extracted transactions for UPDT |
| PAC7MR | &&MR | Output | Extracted transactions for REOR (EXPU) |
| PAC7MX | &&MX | Output | Non extracted entities (PACX) |
| PAC7GY | &&GY | Output | Extracted transactions for UPDP |
| PAC7TD | &&TD | Output | Extracted transactions for CPSN |
| PAC7UE | &&UE | Output | Extracted transactions for EXUE |
| PAC7IA | | Report | General printout of the program stream |
| PAC7DD | | Report | Errors on input transactions |
| PAC7ED | | Report | Extractions report |

| Code | Physical name | Type | Label |
|----------|---------------|--------|--------------------|
| PAC7EE | | Report | Extractions report |
| PAC7EG | | Report | Extractions report |
| PAC7EM | | Report | Extractions report |
| PAC7EP | | Report | Extractions report |
| PAC7EQ | | Report | Extractions report |
| PAC7EU | | Report | Extractions report |
| PAC7EZ | | Report | Extractions report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0: No error
- 4: Error in user input (detailed in PAC7EE) or in EXTR/EXUE extractions (detailed in PAC7EZ)
- 8: Error in '*' line (detailed in PAC7DD) or in EXLI (Database not available)

Deletion of the KSDS work file: IDCAMS

PACX - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**          - EXTRactions FROM DATABASE -
/**          - EXTRactions COMPArator   -
/** -----
/**
/** THE PACX PROCEDURE ALLOWS TO PERFORM VARIOUS TYPES
/** OF DATA EXTRactions FROM THE DEVELOPMENT DATABASE
/** VIA PAF EXTRACTOR.
/**
/** POSSIBLE VALUES FOR THE EXTRACTOR CODE INCLUDE:
/** - EXTR:  EXTRaction OF ENTITIES
/** - EXTA:  EXTRaction OF ENTITIES (EXTRACTED TRANSACTIONS
/**          ARE SORTED, ACCORDING TO THE INPUT
/**          IDENTIFICATION LINES ORDER.
/**          EACH REQUEST IS THUS PRECEDED BY A "*" LINE,
/**          EXTRACTED TRANSACTIONS WILL BE SORTED IN THE
/**          REQUEST ORDER).
/** - EXUE:  EXTRaction OF USER ENTITIES
/** FOLLOWING VALUES ARE RESERVED FOR THE ADMINISTRATOR:

```

```

/** - EXLI:EXTRACTION OF LIBRARIES OR LIBRARY SUB-NETWORKS
/** - EXPJ:EXTRACTION OF JOURNAL (FORMATTING FOR CPSN IS
/**       NOT POSSIBLE)
/** - EXPU:EXTRACTION OF ENTITIES TO BE PURGED
/**       (FORMATTING FOR CPSN IS NOT POSSIBLE)
/** - RMEN:EXTRACTION OF ENTITIES FOR UPLOAD/REPLACEMENT/
/**       RECODING (FORMATTING FOR CPSN IS NOT POSSIBLE).
/**       RMEN IS SUBJECT TO A SEPARATE PURCHASE AGREEMENT
/** - CPSN:COMPARISON OF SUB-NETWORKS.
/**
/** -----
/**
//BVPPACX  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',                INDEX OF SYSTEM NON-VSAM FILES
//          INDUV='$INDUV',                INDEX OF USER VSAM FILES
//          INDUN='$INDUN',                INDEX OF USER NON-VSAM FILES
/**:      VSAMCAT='$VCAT',                USER VSAM CATALOG
/**:      SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',              SORT LIBRARY
//          USER=,                        USER CODE
//          LSR='BLSR',                    LSR BATCH SYSTEM NAME
//          OUT=$OUT,                      UTILITIES AND ERRORS OUTPUT CLASS
//          OUTL=$OUT,                     OUTPUT CLASS OF REPORTS
//          UWK=$UWK,                      WORK UNIT
//          SPAMB='(TRK,(5,1),RLSE)',      REQUEST FILE SPACE
//          SPAMV='(TRK,(50,10),RLSE)',    UPDT FILE SPACE
//          SPAGY='(TRK,(50,10),RLSE)',    UPDP FILE SPACE
//          SPATD='(TRK,(50,10),RLSE)'     CPSN FILE SPACE
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&PACXMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**:      DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY  DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN  DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR

```

```

//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
/*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
/*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//*
//PACX EXEC PGM=BVPACX,REGION=0K
//*-----
/*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
/*:          DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVP GN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVP GR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVP GU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7PJ DD DSN=&INDUN..&BASE.PJ(0),DISP=SHR
//PAC7IA DD SYSOUT=&OUTL
//PAC7DD DD SYSOUT=&OUTL
//PAC7ED DD SYSOUT=&OUTL
//PAC7EE DD SYSOUT=&OUTL
//PAC7EG DD SYSOUT=&OUTL
//PAC7EM DD SYSOUT=&OUTL
//PAC7EP DD SYSOUT=&OUTL
//PAC7EQ DD SYSOUT=&OUTL
//PAC7EU DD SYSOUT=&OUTL

```

```

//PAC7EZ DD SYSOUT=&OUTL
//PAC7MA DD DUMMY
//PAC7ES DD DUMMY
//SYLSR DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//SYSEXT DD SUBSYS=(&LSR,'DDNAME=SYLSR')
//PAC7MB DD DSN=&&PACXMB,DISP=(OLD,DELETE,DELETE)
//PAC7BM DD DSN=&&PACXBM,DISP=(,DELETE),UNIT=&UWK,
// DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7MM DD DSN=&&PACXMM,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=113,BLKSIZE=11300)
//PAC7MJ DD DSN=&&PACXMJ,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=158,BLKSIZE=6320)
//PAC7TE DD DSN=&&PACXTE,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=323,BLKSIZE=6460)
//PAC7RE DD DSN=&&PACXRE,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=36,BLKSIZE=6012)
//PAC7RM DD DSN=&&PACXRM,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7RQ DD DUMMY
//PAC7WD DD DSN=&&PACXWD,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7MV DD DSN=&&MV,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MR DD DSN=&&MR,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MX DD DSN=&&MX,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7TD DD DSN=&&TD,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7GY DD DSN=&&GY,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAGY,
// DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)
//PAC7UE DD DSN=&&UE,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=230,BLKSIZE=6440)
//SYSUDUMP DD SYSOUT=&OUT
//*
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR

```

```
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSEXT),DISP=SHR
//PACROU DD DSN=&&DLSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSEXT,DISP=(OLD,DELETE)
```

Chapter 4. Personalized Extraction/Automated Documentation

Foreword

The PAF+/Extraction and the PDM+/Outline functions can be used separately or together.

PAF+ is used to write the Extraction Master Path and execute it when the PTE_x is a User Extractor.

PDM+ is used to write and execute the Master Outline (PT_Ed).

The PAF-PDM functions are used when the Master outline calls an Extraction Master Path of the Macro-Command type.

- If you use the PAF+/Extraction function alone, you can generate User Extractor programs and possibly format the extracted data.
- If you use the PDM+/Outline function alone, you can create skeletons to standardize the printing of Volumes (standard Print Options, Text instances always called, standardized calls).
- If you use both functions together, PAF+ extracts data from the Database. This data is processed by PDM+ and finally printed in a Volume.

For more information on these functions, refer to the 'Pacbase Access Facility (PAF)' and the 'Personalized Documentation Manager (PDM)' manuals.

Personalized Extractions - PAF+

XPAF - Validation of an Extraction Master Path

XPAF - Introduction

The Extraction Master Path validation procedure, XPAF, is used to perform specific extractions that the standard procedures cannot perform. See the 'Pacbase Access Facility (PAF)' reference manual.

Results

The type of result depends on whether or not the extracted domain is to be integrated into a Document : Macro-Command or User Extraction program.

A Macro-Command is a subroutine to be activated in a GPRT print request (choice: PCV).

A User Extraction program is a Source Program to be compiled and executed.

Prerequisite

In order to use this procedure, the Database Manager must update the Database with the transaction file, supplied for installation, which contains the .PPTEx extension Meta-Entity, whose type is 7E (VINS procedure).

Implementation

Before the procedure can be executed, the user must define an instance of this extension meta-entity (Y7E). Its Definition and Description determine the characteristics and format of the general extraction program.

Abnormal execution

Whatever the cause of the abend, the procedure can be re-executed once the problem has been solved.

Printed output

This procedure prints a validation report and a simulation of the Extraction Master Path.

XPAF - User Input

One '*' line per library and session to be consulted

| Position | Length | Value | Meaning |
|----------|--------|----------|-----------------|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |
| 19 | 3 | bbb | Library code |
| 22 | 4 | nnnn | Session number |
| 26 | 1 | 'T' | Session version |
| 68 | 1 | ' ' | Standard print |
| | | '1' | Uppercase print |

One command line 'EX' for the following elements:

| Position | Length | Value | Meaning |
|----------|--------|----------|--|
| 2 | 2 | 'EX' | Line code |
| 4 | 2 | | Extension ME Type (Y7E by default) |
| 6 | 6 | eeeeee | User Entity code |
| | | | Warning: Specify library and session if the MEs whose instances are to be extracted are in a parallel sub-network (UEs extractions managed by the WorkStation for example) |
| 12 | 3 | bbb | Library code |
| 15 | 4 | nnnn | Session number |
| 19 | 1 | 'T' | Session version |
| 20 | 6 | 'UPDATE' | Update of GS |
| | | SPACE | Check of the presence of the Master Path in GS (no update). Check of the user entity's use in the sub-network. |

Examples

*user passwordLIB

EX7EEXT001_____UPDATE

*user passwordLIB

EX7EEXT002

XPAF - Description of Steps

Input recognition: PTU001

Access and validation: PTEX30

| Code | Physical name | Type | Label |
|---------|----------------------|-------|-----------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACG GN | &INDSV..BVP GN | Input | Administration Database Index |
| PACG GR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACG GU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AN | &INDUV.. &BASE.AN | Input | Development Database Index file |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |

| Code | Physical name | Type | Label |
|----------|-----------------|----------------|--------------------------------------|
| PAC7AY | &INDUV.&BASE.AY | Input | Development Database Extension Data |
| PAC7MB | &&XPAFMB | Input | User Input |
| PAC7SP | &INDSV..BVPSP | Input | Variable skeleton file |
| PAC7GS | &INDSV..BVPGS | Input / Output | Extraction Paths |
| PAC7ED | &&PAC7ED | Output | Report passed on to printing program |
| PAC7GP | &&PAC7GP | Output | Temporary generated source |
| PAC7DD | | Report | Report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

.Return code(s):

- 0: OK
- 8: Error on Database or no authorization for Batch procedure : no extraction

Extractor generation: PTEX80

| Code | Physical name | Type | Label |
|--------|---------------|--------|-----------------------------------|
| PAC7AE | &INDSV..BVPAE | Input | Error Messages |
| PAC7SF | &INDSN..BVPSF | Input | Fixed skeleton file |
| PAC7GP | &&PAC7GP | Input | Source file generated by PTEX30 |
| PAC7ST | &&PPAFMB | Output | Generated source to be translated |

Preprocessor: PAFP10

| Code | Physical name | Type | Label |
|--------|-----------------|--------|-----------------------------------|
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV.&BASE.AN | Input | Development Database Index file |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAF80 | &PAFINPUT | Input | Generated programs |
| COB80 | &&PAFGEN | Output | Generated programs to be compiled |

| Code | Physical name | Type | Label |
|--------|---------------|--------|--------------|
| PAFREP | | Report | Error report |

PTEX printing: PTEXD0

| Code | Physical name | Type | Label |
|----------|----------------|------------------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7ED | &&PAC7ED | Input | PTEX30 Report |
| PAC7GS | &INDSV..BVP GS | Input/ Output | Extraction Paths |
| PAC7RD | | Report | Control report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

XPAF - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**              - XPAF (PAF EXTENSION) -
/**
/** -----
/**
/** THE EXTRACTION MASTER PATH VALIDATION PROCEDURE,
/** XPAF, ALLOWS FOR THE SIMULATION OF SPECIFIC EXTRACTIONS
/** THAT THE STANDARD PROCEDURES ARE NOT ABLE TO PERFORM.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/**   COL 2 : "*"
/**   COL 3 : USERIDXX
/**   COL 11 : PASSWORD
/**   COL 19 : (BBB)   LIBRARY CODE
/**   COL 22 : (4 N)   SESSION NUMBER
/**   COL 26 : (1 CAR.) SESSION VERSION
/**   COL 68 : " "     STANDARD PRINT
/**             "1"    UPPERCASE PRINT
/** - COMMAND LINE :
/** COL 2 : "EX"      LINE CODE
/** COL 4 : (2 CAR.)  METAENTITY TYPE (7E BY DEFAULT)
/** COL 6 : (6 CAR.)  USER ENTITY CODE

```

```

/** COL 12 : (BBB) LIBRARY CODE (IF THE U.E.O.
/** COL 15 : (4 N) SESSION NUMBER ARE IN PARALLEL
/** COL 19 : (1 CAR.) SESSION VERSION SUB-NETWORK)
/** COL 20 : "UPDATE" UPDATE OF GS
/** " " CHECK OF THE PRESENCE OF THE
/** MASTER PATH IN GS.
/**
/** -----
/**
/**BVPXPAF PROC BASE=$BASE, CODE OF DEVPT DATABASE
/** INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
/** INDSN='$INDSN', INDEX OF SYSTEM NON-VSAM FILES
/** INDUV='$INDUV', INDEX OF USER VSAM FILES
/** STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/** SORTLIB='$BIBT', SORT LIBRARY
/**: VSAMCAT='$VCAT', USER VSAM CATALOG
/**: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
/** LSR='BLSR', LSR BATCH SYSTEM NAME
/** OUT=$OUT, UTILITIES AND ERRORS OUTPUT CLASS
/** UWK=$UWK, WORK UNIT
/** SPAGP='(TRK,(100,10))', GENERATION FILE SPACE
/** SPAED='(TRK,(100,10),RLSE)', SPACE OF PRINTING FILES
/** CYL='(3,1)' SORTWORK SPACE
/*******
/**INPUT EXEC PGM=BVPTU001
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/** DD DSN=$BCOB,DISP=SHR
/**CARTE DD DDNAME=SYSIN
/**PAC7MB DD DSN=&XPAFMB,DISP=(,PASS),
/** UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
/** DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**PTEX30 EXEC PGM=BVPTEX30,REGION=0K
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/** DD DSN=$BCOB,DISP=SHR
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**: DD DSN=&SYSTCAT,DISP=SHR
/**SYSOUT DD SYSOUT=&OUT
/**SYSUDUMP DD SYSOUT=&OUT
/**ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
/**PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
/**ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
/**PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
/**AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
/**PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
/**PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
/**GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
/**PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
/**GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
/**PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
/**GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
/**PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
/**PAC7SP DD DSN=&INDSV..BVPSP,DISP=SHR
/**PAC7DD DD SYSOUT=&OUT

```

```

//PAC7MB DD DSN=&&XPAFMB,DISP=(OLD,DELETE,DELETE)
//PAC7ED DD DSN=&&PAC7ED,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=144,BLKSIZE=1440),
// SPACE=&SPAED
//PAC7GP DD DSN=&&PAC7GP,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=119,BLKSIZE=1190),
// SPACE=&SPAGP
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PTEX80 EXEC PGM=BVPTEX80,REGION=0K,COND=(8,LE,PTEX30)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7GP DD DSN=&&PAC7GP,DISP=(OLD,PASS,DELETE)
//PAC7SF DD DSN=&INDSN..BVPSF,DISP=SHR
//PAC7ST DD DSN=&&PPAFMB,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
// SPACE=&SPAGP
//PAFP10 EXEC PGM=BVPAFP10,COND=(8,LE,PTEX30)
//*****
//* - PAF PRE PROCESSING *
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAF80 DD DSN=&&PPAFMB,DISP=(OLD,DELETE,DELETE)
//COB80 DD DSN=&&PAFGEN,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
// SPACE=&SPAGP
//PAFREP DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PTEXD0 EXEC PGM=BVPTEXD0,REGION=0K,COND=(12,LE,PAFP10)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')

```

```

//PAC7ED DD DSN=&&PAC7ED,DISP=(OLD,PASS,DELETE)
//PAC7RD DD SYSOUT=&OUT
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

XPAF - Operation of an Extraction Master Path

Execution of a user extractor (E-type PTE_x)

Once validated, compiled, and linked, a User Extractor is ready for execution.

Execution of a macro-command (M-type PTE_x):

Once validated, compiled, and linked, a Macro-Command is not ready for execution. It must be called in a Master Outline.

Note

An Extraction Master Path is independent of the Database in which it is defined and described.

Documentation Structuring - PDM+

XPDM - Validation of a Master Outline

XPDM - Introduction

A Master Outline is a P-type Volume ('V' entity) designed to be called in another PDM Volume. Its functions are to:

- Memorize general descriptions (print option, for example) so that they do not have to be redefined in each Volume.
- Print the information extracted via an Extraction Master Path. This function may be recursive.

If no serious error is detected, the XPDM procedure updates the Extraction Master Path file (GS). It can also be used without updating the GS file.

See the 'Personalized Documentation Manager' manual for more details.

Abnormal execution

Whatever the cause of the abend, the procedure can be re-executed once the problem has been solved.

Printed output

This procedure prints the description of a Master Outline, as well as the comments, and a list of the anomalies found, if any.

XPDM - User Input

One '*' line to define the context.

| Position | Length | Value | Meaning |
|----------|--------|----------|-----------------|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |
| 19 | 3 | bbb | Library code |
| 22 | 4 | nnnn | Session number |
| 26 | 1 | 'T' | Session version |
| 68 | 1 | ' ' | Standard print |
| | | '1' | Uppercase print |

One 'EP' line for the following elements :

| Position | Length | Value | Meaning |
|----------|--------|----------|---|
| 2 | 2 | 'EP' | Line code |
| 4 | 6 | vvvvvv | Volume code |
| 10 | 6 | 'UPDATE' | GS file update |
| | | SPACE | Check of the volume's presence in GS Check of the volume's use in the sub-network. No GS file update if presence or use. |

Examples

*user passwordLIB

EPMANUALUPDATE

*user passwordLIB

EPMANUAL

XPDM - Description of Steps

Input recognition: PTU001

Extraction of master outline: PTED30

| Code | Physical name | Type | Label |
|--------|------------------|------------------|-------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AY | &INDUV..&BASE.AY | Input | Development Database Extension data |
| PAC7MB | &&XPDM MB | Input | User Input |
| PAC7GS | &INDSV..BVP GS | Input/ Output | Extraction Paths |
| PAC7ED | &&PAC7ED | Output | File for report |
| PAC7SG | &&PAC7SG | Output | GS-Update preparation |
| PAC7DD | | Report | Report |

GS update and printing of the master outline: PTED60

| Code | Physical name | Type | Label |
|----------|----------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7ED | &&PAC7ED | Input | Print file |
| PAC7SG | &&PAC7SG | Input | GS Update preparation file |
| PAC7GS | &INDSV..BVP GS | Output | Extraction Paths |
| ETATGP | | Report | Output report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

XPDM - Execution JCL

```
/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**          - XPDM  (PDM EXTENSION) -
/**
/** -----
/**
/** A MASTER OUTLINE IS A P-TYPE VOLUME ("V" ENTITY)
/** DESIGNED TO BE CALLED IN ANOTHER PDM VOLUME.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/**   COL 2  : "*"
/**   COL 3  : USERIDXX
/**   COL 11 : PASSWORD
/**   COL 19 : (BBB)   LIBRARY CODE
/**   COL 22 : (4 N)   SESSION NUMBER
/**   COL 26 : (1 CAR.) SESSION VERSION
/**   COL 68 : " "     STANDARD PRINT
/**           "1"     UPPERCASE PRINT
/** - COMMAND LINE :
/** COL 2  : "EP"     LINE CODE
/** COL 4  : (6 CAR.) REPORT CODE
/** COL 10 : "UPDATE" UPDATE OF GS
/**           " "     CHECK OF THE PRESENCE OF VOLUME
/**           " "     IN GS.
/**
/** -----
/**
/**BVPXPDM  PROC BASE=$BASE,           CODE OF DEVPT DATABASE
/**          INDSV='$INDSV',           INDEX OF SYSTEM VSAM FILES
/**          INDUV='$INDUV',           INDEX OF USER VSAM FILES
/**          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/**          SORTLIB='$BIBT',           SORT LIBRARY
/***:       VSAMCAT='$VCAT',           USER VSAM CATALOG
/***:       SYSCAT='$SCAT',           SYSTEM VSAM CATALOG
/**          LSR='BLSR',               LSR BATCH SYSTEM NAME
/**          OUT=$OUT,                 UTILITIES AND ERRORS OUTPUT CLASS
/**          UWK=$UWK,                 WORK UNIT
/**          SPASG='(TRK,(2,1))',      EXTRACTION SCHEMAS FILE SPACE
/**          SPAED='(TRK,(5,1),RLSE)', SPACE OF PRINTING FILES
/**          CYL='(3,1)'                SORTWORK SPACE
/*******
/**INPUT  EXEC PGM=BVPTU001
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**        DD DSN=$BCOB,DISP=SHR
/**CARTE   DD DDNAME=SYSIN
/**PAC7MB  DD DSN=&XPDMMB,DISP=(,PASS),
/**          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
/**          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**PTED30 EXEC PGM=BVPTED30,REGION=0K
/**-----
```

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&XPDMMB,DISP=(OLD,DELETE,DELETE)
//PAC7ED DD DSN=&&PAC7ED,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=203,BLKSIZE=2030),
// SPACE=&SPAED
//PAC7SG DD DSN=&&PAC7SG,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=214,BLKSIZE=27392),
// SPACE=&SPASG
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//PTED60 EXEC PGM=BVPTED60,REGION=0K
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7ED DD DSN=&&PAC7ED,DISP=(OLD,DELETE,DELETE)
//PAC7SG DD DSN=&&PAC7SG,DISP=(OLD,DELETE,DELETE)
//PAC7GS DD DSN=&INDSV..BVPGS,DISP=SHR
//ETATGP DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

Extraction Master Path and Outline File

PRGS - Printing of Master Path / Outline File

PRGS - Introduction

The PRGS procedure prints the content of the PAC7GS file, where the Master Outlines and Extraction Master Paths are stored.

Result

A printout showing the Extraction Master Path and the associated Master Outlines.

PRGS - User Input

One '*' line to identify the user.

| Position | Length | Value | Meaning |
|----------|--------|----------|---------------|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |

PRGS - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Printing of the master path and outline file: PTEP90

| Code | Physical name | Type | Label |
|----------|----------------|--------|-------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7GS | &INDSV..BVP GS | Input | Extraction Paths |
| PAC7MB | &&PRGSMB | Input | User Input |
| PAC7DD | | Report | Output Report |
| ETATGS | | Report | Master Path and Outline file report |
| SORTWK01 | | Sort | |

| Code | Physical name | Type | Label |
|----------|---------------|------|-------|
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

PRGS - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - PRINT OF MASTER PATH FILE -
/**
/** -----
/**
/** THE PRGS PROCEDURE PRINTS THE CONTENTS OF THE
/** PAC7GS FILE, WHERE MASTER OUTLINES AND EXTRACTION
/** MASTER PATHS ARE STORED.
/**
/** -----
/**
/**BVPPRGS  PROC INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
/**      INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
/**      STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
/**      SORTLIB='$BIBT',          SORT LIBRARY
/***:      VSAMCAT='$VCAT',          USER VSAM CATALOG
/***:      SYSTCAT='$SCAT',          SYSTEM VSAM CATALOG
/**      OUT=$OUT,          OUTPUT CLASS
/**      UWK=$UWK,          WORK UNIT
/**      CYL='(3,1)'          SORTWORK SIZE
/*******
/**INPUT EXEC PGM=BVPTU001
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**      DD DSN=$BCOB,DISP=SHR
/**CARTE DD DDNAME=SYSIN
/**PAC7MB DD DSN=&&PRGSMB,DISP=(,PASS),
/**      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
/**      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**VERIFY EXEC PGM=IDCAMS
/**-----
/***:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/***:      DD DSN=&SYSTCAT,DISP=SHR
/**SYSPRINT DD SYSOUT=&OUT
/**PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
/**PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
/**PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
/**SYSIN DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
/**      DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
/**      DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
/**PTEP90 EXEC PGM=BVPTEP90,REGION=0K
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**      DD DSN=$BCOB,DISP=SHR

```

```
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//PACGGN DD DSN=&INDSV..BVP GN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVP GR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7GS DD DSN=&INDSV..BVP GS,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&PRGSMB,DISP=(OLD,DELETE,DELETE)
//ETATGS DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
```

Chapter 5. Batch Update

UPDP - Update from PAF Tables

UPDP - Introduction

The UPDP procedure performs an update of the Database from a sequential file which is the image of PAF tables.

The operating principle of UPDP is very similar to that of UPDT, with the exception that input transactions have a different format.

Abnormal execution

Refer to the 'Abnormal Execution' section of the UPDT procedure.

UPDP - User Input / Update Rules / Results

The sequential file of input transactions is produced by a PAF extractor program or by the PACX procedure. Its records reflect the PAF tables format. For a detailed description of these tables, see the documentation of the Pacbase Access Facility Tables.

| Position | Length | Meaning |
|----------|--------|--|
| 1 | 1 | Transaction code (C, M, X, A or D, B, S) |
| 2 | 10 | PAF table code |
| 12 | 299 | PAF table contents (as described in the PAF Tables Reference Manual) |

There are restrictions on the Client and Extension User Entities Definition and Description tables.

The size of the UPDP input file is 310 characters long while the size of these tables exceeds 310 characters. The records must then be re-formatted in the following manner:

Client and Extension User Entities Definition Tables - \$TTDEF and YTTDEF.

| Position | Length | Meaning |
|----------|--------|--|
| 1 | 1 | Transaction code (C, M, X, A or D, B, S) |
| 2 | 10 | Table code |

| Position | Length | Meaning |
|----------|--------|---|
| 12 | 1 | Record continuation code: blank character for the first record, any character for the continuation records. |
| 13 | 1 | Separator (1) |
| 14 | 55 | Explicit keywords |
| 69 | 305 | Contents of the concatenated columns |

Client and Extension User Entities Description tables - \$TTDxx and YTTDxx.

| Position | Length | Meaning |
|----------|--------|--|
| 1 | 1 | Transaction code (C, M, X, A or D, B) |
| 2 | 10 | Table code |
| 12 | 1 | Record continuation code: blank character for the first record, any character for the continuation records |
| 13 | 1 | Separator (1) |
| 14 | 30 | User Entity code |
| 44 | 305 | Contents of the concatenated columns |

(1) If the separator is set to blank, data items are concatenated one after the other ; the length of each data item is respected.

If the separator is specified, data items are separated with this character. The non-significant blanks located at the end of the data item are optional.

The separator must not be present in the data item.

Update rules

Update transactions are not sorted.

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

| Position | Length | Value | Meaning |
|----------|--------|----------|----------------|
| 2 | 10 | 'ASSIGN' | Table code |
| 12 | 8 | uuuuuuuu | User code |
| 20 | 8 | pppppppp | Password |
| 28 | 3 | bbb | Library code |
| 31 | 4 | ssss | Session number |

| Position | Length | Value | Meaning |
|----------|--------|--------|---|
| | | ' ' | current session |
| 35 | 1 | 'T' | Session status: Test session |
| 40 | 3 | ppp | Product code (in case of a Database under DSMS control) |
| 43 | 6 | nnnnnn | Product number (in case of a Database under DSMS control) |
| 49 | 1 | | Top generated by the extractor |
| | | ' ' | Manual transactions |
| | | 'N' | Extractor transactions |
| | | 'G' | Transactions output from the retrieval of PG25 or PP25 |

When the update is performed while the on line mode is active (on platforms that support this functionality), the input transaction flow must be preceded by a CHECKP table code line.

| Position | Length | Value | Meaning |
|----------|--------|----------|--|
| 2 | 10 | 'CHECKP' | Table code |
| 12 | 4 | nnnn | Number of transactions processed between two pauses or checkpoints |
| 16 | 4 | 'UPDT' | Update procedure |
| 20 | 2 | nn | LAN Platforms: Pause time, in seconds, between two update sets |

Printed output

Refer to the description of the UPDT output.

Result

Refer to the description of the UPDT result.

UPDP - Description of Steps

Check of VSAM files: IDCAMS

Database consistency check: PTUBAS

| Code | Physical name | Type | Label |
|--------|-----------------|-------|---------------------------|
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database data |

| Code | Physical name | Type | Label |
|--------|---------------|--------|-----------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAC7DS | | Report | Validity report |

Return code:

This utility sends a return code 12 and the procedure stops running if the database is invalid.

Transaction formatting: PAF900

| Code | Physical name | Type | Label |
|--------|----------------------|--------|--|
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV.. &BASE.AN | Input | Development Database Index file |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7GY | &PAFINPUT | Input | Update transactions |
| PAC7MV | &&PAC7MV | Output | Formatted transactions (must be able to contain all input transactions as well as elementary deletion transactions generated by multiple deletion transactions) (length=170) |
| PAC7ME | &&PAC7ME | Output | Work file (length=372) |
| PAC7MW | | Output | Work file (length=170) |
| PAC7MX | | Output | Work file (length=743) |
| PAC7MY | | Output | Work file (length=743) |

Update of the Development Database: PACA15

| Code | Physical name | Type | Label |
|--------|----------------------|--------|---------------------------------|
| PAC7AR | &INDUV..&BASE.AR | Output | Development Database Data file |
| PAC7AN | &INDUV.. &BASE.AN | Output | Development Database Index file |
| PAC7AY | &INDUV..&BASE.AY | Output | Development Database Extension |

| Code | Physical name | Type | Label |
|--------|------------------|--------|---|
| PAC7AJ | &INDUV..&BASE.AJ | Output | Development Database Journal |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGY | &INDSV..BVPGY | Input | Administration Database Extension |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7DC | &DSMS | Input | Development Database elements DSMS file |
| PAC7ME | &&PAC7ME | Input | Work file |
| PAC7MV | &&PAC7MV | Input | Update transactions |
| PAC7RB | &&PAC7RB | Output | UPDT erroneous transactions (length=80) |
| PAC7RY | &&PAC7RY | Output | UPDP erroneous transactions (length=310) |
| PAC7IE | | Report | Update report (length=132) |
| PAC7IF | | Report | List of erroneous transactions (length=132) |

The list of transactions specific to a user is preceded by a banner with this user's code.

Return codes :

- 0 : OK without error
- 2 : Warning
- 4 : Error

UPDP - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - BATCH UPDATE FROM PAF TABLES -
/**
/** -----
/**
/** THE UPDP PROCEDURE PERFORMS AN UPDATE OF THE DATABASE
/** FROM A SEQUENTIAL FILE REFLECTING PAF TABLES.
/**
/** THE SEQUENTIAL FILE OF INPUT TRANSACTIONS IS PRODUCED

```

```

/** BY A PAF EXTRACTOR PROGRAM. ITS RECORDS MIRROR
/** THE PAF TABLES.
/** EACH SET OF TRANSACTIONS IMPACTING A LIBRARY OR SESSION
/** MUST BE PRECEDED BY AN ASSIGN TABLE CODE LINE.
/** WHEN THE UPDATE IS PERFORMED WHILE THE TP IS ACTIVE
/** (ON PLATFORMS THAT SUPPORT THIS FUNCTIONALITY),
/** THE INPUT TRANSACTION FLOW MUST BE PRECEDED BY A CHECKP
/** TABLE CODE LINE.
/** -----
/**
//BVPUPDP  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES
//          INDSN='$INDSN',                INDEX OF NON-VSAM SYSTEM FILES
//          INDUV='$INDUV',                INDEX OF VSAM USER FILES
//          PAFINPUT=,                    DSNAME OF PAF INPUT TABLES
//          DSMS='$DSMS',                DSNAME OF PRODUCT ELEMENTS DSMS FILE
/**:       VSAMCAT='$VCAT',                USER VSAM CATALOG
/**:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
//          LSR='BLSR',                    LSR BATCH SYSTEM NAME
//          OUT=$OUT,                      EXECUTION ERRORS OUTPUT CLASS
//          OUTL=$OUT,                      OUTPUT CLASS OF REPORTS
//          UWK=$UWK,                       WORK UNIT
//          SPAMB='(TRK,(100,10),RLSE)'    TRANSACTION SPACE
//*****
//VERIFY EXEC PGM=IDCAMS
/**-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTUBAS   EXEC PGM=BVPTUBAS
/**-----
/**:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
/**:       DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7DS   DD SYSOUT=&OUT

```

```

//PAF900 EXEC PGM=BVPAF900,COND=(0,NE,PTUBAS),REGION=0K
//*-----
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY DD DSN=&PAFINPUT,DISP=SHR
//PAC7ME DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=372,BLKSIZE=5952)
//PAC7MV DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW DD DISP=(,DELETE),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX DD DISP=(,DELETE),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY DD DISP=(,DELETE),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA15 EXEC PGM=BVPACA15,COND=((0,NE,PTUBAS),(0,NE,PAF900))
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR

```

```

//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC DD DSN=&DSMS,DISP=SHR
//PAC7IE DD SYSOUT=&OUTL
//PAC7IF DD SYSOUT=&OUTL
//PAC7ME DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB DD DSN=&&PAC7RB,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PAC7RY DD DSN=&&PAC7RY,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)
//SYSUDUMP DD SYSOUT=&OUT

```

UPDT - Update

UPDT - Introduction

The Database update procedure (UPDT) executes a batch update of the Database. It allows access to all the libraries according to the authorizations of the different users.

With the DSMS facility (DSM), this procedure reads the VisualAge Pacbase elements file (DC).

Execution conditions

Since the Database is updated, the AR, AN, AJ and AY files must be closed to on-line use, except for hardware environments that support concurrent on-line and batch access.

Note

For very large updates (in terms of number of transactions, about 5000), before executing this procedure, it may be necessary to:

- Save, archive and restore the Database to increase the space allocated to the files or to physically reorganize the files in order to make all the free space initially provided available.
- Temporarily suppress Journalization
(See chapter 'Database Management Utilities', subchapter 'Database Restoration', in 'The Administrator's Procedures' manual').

This procedure updates the current session number in two cases:

- When it is the first connection of the day to the Database, and

- When it contains a Database Freeze request submitted by the Administrator (see 'The Administrator's Procedures' manual)

Abnormal execution

Refer to chapter 'Overview', subchapter 'Abnormal Endings' in 'the Administrator's Procedures' manual.

There are two types of abnormal executions:

- Abnormal execution which occurs before the execution of the BVPACA15 program, or during the opening of files in this program. The procedure can be restarted after the problem is corrected.
- Abnormal execution which occurs during the execution of the BVPACA15 program. The Database is left in an inconsistent state if there is no rollback. If the problem appeared during input-output on a Database file, the printed error message and the file status will lead you to the solution.

If the procedure execution has stopped with the error 'Short label already used', the Database remains consistent and the procedure can be restarted after the label has been corrected.

In either case, you can re-start the procedure only by using a backup file and apply the archived transactions subsequent to this backup (REST procedure).

UPDT - User Input / Update Rules / Results

Update rules

Each set of transactions impacting a Library must be preceded by a *-type line which specifies the context.

These transactions are not sorted.

Printed output

The two printed output generated by this procedure are:

- A global report on the update,
- A list of the rejected update transactions.

They are printed by the user, and the transaction groups are separated by a flag.

Result

Output of the UPDT procedure is:

- A Database ready to be used in on-line or batch mode.

- A Journal file of the transactions that have modified the Database (as long as there was no inhibit request during the last restoration).

These transactions are made up of a common part which contains the action code, a line identifier and a specific part which is detailed in the following sections for each Description of entity.

Action codes

| Action code | Label |
|-------------|--|
| C | Creation of a line in the library |
| M | Modification of a line. |
| Blank | Creation or modification of a line, depending on its presence or absence in the library. |
| X | Creation or modification with possible use of ampersand (&). |
| D | Deletion of one line. |
| B | Multiple lines deletion, starting with this line. |
| R | End of multiple lines deletion up to and including this line. |
| S | Complete deletion of an entity |

Note about deletions

If an entity is used in several Libraries, deletions in a lower Library are rejected.

It is possible to globally delete (using ACTION CODE 'B') an entity and all of its uses in Screens, Reports or Segments. However, these deletions will be effective only in update Libraries.

The B code generates elementary deletion transactions.

The S code can be used on an entity definition only, one transaction only will be journalized. Checks will be done before the update.

Caution

A field which is not valued is not modified. Enter the '&' character to reset this field to blank.

Specific action codes: 'F' and 'P'

The 'F' and 'P' action codes are used in extractions for updates.

The 'F' value is used to force an update, i.e. after an extraction (via EXLI or any other extractor), it allows the creation of an incomplete Definition so that the X-references to these entities (usually User Entities) can be satisfied, a sort being impossible.

This code triggers the update of the Database.

The 'P' value allows an identification line to be assigned to all the Description lines that follow without updating the Definition of this entity (e.g. 'P' lines of a Program in a Library where the Definition exists only in a higher Library).

Multi-entity User Input

Multi-purpose Line (Line VC, VG, VE, VO):

The VC-code line is used for calling Parameterized Input Aids and for assigning Comments to an Entity or some description lines.

Insertion of comments (VC line)

- a line which contains the type and code of the concerned entity and the line number,
- a line which contains the comments in column 4 and the character '*' in column 80 (value for continuation line).

Call of an Input Aid (VC and VZ lines)

- one VC line only is needed. It contains the type and code of the entity concerned, the line number if it is a Description and the value 'T' in the type of line column as well as the code of the Input Aid.
- a VZ line per variable part of the called Input Aid (see the following section, Parameterized Input Parameterized Input Aids/Variable Parts), the line subnumber and the description value.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'VC' | Line code for a 'GC' screen |
| | | 'VG' | Line code for a 'GG' screen |
| | | 'VE' | Line code for a 'GE' screen (call of a P.I.A. not possible in this screen) |
| | | 'VO' | Line code for a 'GO' screen |
| 4 | 2 | | Entity type receiving the Comments |
| 6 | 30 | | Entity code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 36 | 3 | | Line number |
| 39 | 3 | | Number of the commented line |
| 42 | 1 | | Line type |
| 43 | 6 | | Code of called P.I.A. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'VC' | Line code for a 'GC' screen |
| | | 'VG' | Line code for a 'GG' screen |
| | | 'VE' | Line code for a 'GE' screen (call of a P.I.A. not possible in this screen) |
| | | 'VO' | Line code for a 'GO' screen |
| 4 | 60 | | Comment line |
| 80 | 1 | | Continuation line |
| | | '*' | This value must be entered to indicate a continuation line. |

Parameterized Input Aids/Variable Parts (Line VZ):

The access line used for entering the contents of the variable parts is 'VZ'.

The structure of the VZ line must copy the P.I.A.'s Description one. The variable parts follow each other. There are no delimiters. The resolution includes the maximum length of each parameter defined.

Note

This line code always comes after a VC line (call of P.I.A.).

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'VZ' | |
| 4 | 2 | | Number of parameter cards in a P.I.A |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 6 | 20 | | Printed label for level n |
| | | | This field contains the fixed part of a P.I.A. line as displayed when the P.I.A. is called. |
| | | | Its contents depend upon the TYPE OF P.I.A. LINE. |
| | | | The label is not justified (to be next to the value, it must be right-justified). |
| | | | On P.I.A. lines to be generated (value 'G' in the LINE GENERATION OPTION field on the P.I.A. Description (-D) screen), each instruction must be left-justified, and, if it does not fit on a single line, its continuation must begin with at least one 'blank' character. |
| 26 | 40 | | DESCRIPTION / SECOND PART |
| | | | This field is specific to a P.I.A. call. |
| | | | With value 'C2' in the OPERATION CODE field, the cursor automatically tabs to the first position of this field. |
| | | | This field is initialized with blanks (default value) or with the value specified in the INITIAL VALUE field for a Standard PIA description line (Type = 'blank'). |
| | | | If symbolic parameters have been defined on the P.I.A. Description (-D), they may be entered in this field. They will be replaced by their corresponding value, and will remain displayed on the right of the screen. |

Call of Instances via Relations (Line QR):

The access line used for the call of instances via Relations is 'QR'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'QR' | |
| 4 | 2 | | Entity type receiving the Comments |
| 6 | 2 | | Meta Entity Type |
| | | | It is an alphanumeric code entered upon creation and which characterizes the Meta Entity in all its uses (two Meta Entities cannot have the same type). |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | The type cannot be modified if User Entities have already been defined for this ME. |
| | | | When used to define or describe a User Entity, this type is preceded by the '\$' character (example: if the 'JOB' ME type is 'JO', the User Entities are referenced by '\$JO.....'). |
| 8 | 30 | | Entity code (30 characters) |
| 38 | 3 | | Line number |
| | | | Numeric. You are advised to begin with line number '100' and then number them in intervals of 20. This facilitates subsequent line insertions, as necessary. |
| | | | This field is alphanumeric if you generate a customized SQL access. In this case, you can enter letters in the 'LIN' field. You can then create more than the '1000' lines initially available. |
| 41 | 6 | | User Relation code |
| 47 | 30 | | Code of called entity (30 charac.) |

Entity Update Lock (Line R):

The access line used to lock the update of entities is 'R'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'R' | |
| 3 | 2 | | Entity type |
| | | | This field is used to specify the type of entity to which one or more keywords are assigned. |
| | | 'K1' | Model Entity. |
| | | 'S' | Text. |
| | | 'C' | Data Element. |
| | | 'A' | Data Structure. |
| | | '2' | Segment. |
| | | 'V1' | Parameterized Input Aid. |
| | | 'L1' | Database Block. |
| | | 'H' | Screen. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | 'B' | Report. |
| | | '0' | Program. |
| | | 'U' | User Manual. |
| | | 'W1' | Volume. |
| | | 'Y1' | User Entity. |
| | | 'Y3' | Client Meta-Entity. |
| | | 'tt' | tt User Entity. Used for updating keywords of tt User Entities. |
| | | 'Y5' | User-Defined Relation. |
| 5 | 2 | | Meta Entity Type |
| | | | It is an alphanumeric code entered upon creation and which characterizes the Meta Entity in all its uses (two Meta Entities cannot have the same type). |
| | | | The type cannot be modified if User Entities have already been defined for this ME. |
| | | | When used to define or describe a User Entity, this type is preceded by the '\$' character (example: if the 'JOB' ME type is 'JO', the User Entities are referenced by '\$JO.....'). |
| 7 | 30 | | Entity code |
| 37 | 36 | | Entity name/comments |
| 73 | 8 | | User code |

Search by Keywords (Line G):

'G' is the access line used to define and assign explicit keywords.

On a first line, you find the type and code of the entity concerned.

Keywords (55 characters) are entered on a second line, a continuation line (identified by the '*' character at the end of the line).

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'G ' | |
| 4 | 2 | | Entity type receiving the Comments |
| 6 | 30 | | Entity code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 36 | 1 | | Call type |
| | | '\$' | Used to update keywords for User Entities. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'G ' | |
| 4 | 55 | | Keywords |
| 80 | 1 | | Continuation line |
| | | '*' | This value must be entered to indicate a continuation line. |

Data Elements

Definition (Line C):

'C' is the access line used to define an Element.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'C' | |
| 3 | 6 | | Element code |
| 9 | 1 | | Type (property, element or alias) |
| 10 | 36 | | Data Element name |
| 46 | 1 | | Type of format |
| | | 'I' | Internal format. |
| 47 | 10 | | Data Element internal format |
| 74 | 1 | | Element internal use |
| 75 | 6 | | Code of parent Data Element |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 2 | 1 | | Line code |
| | | 'C' | |
| 3 | 6 | | Element code |
| 9 | 1 | | Type (property, element or alias) |
| 10 | 36 | | Data Element name |
| 46 | 1 | | Type of format |
| | | 'E' | Input format. |
| 47 | 10 | | Conversational format |
| 74 | 1 | | Element internal use |
| 75 | 6 | | Code of parent Data Element |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'C' | |
| 3 | 6 | | Element code |
| 9 | 1 | | Type (property, element or alias) |
| 10 | 36 | | Data Element name |
| 46 | 1 | | Type of format |
| | | 'S' | Output format. |
| 47 | 27 | | Output Format |
| 74 | 1 | | Element internal use |
| 75 | 6 | | Code of parent Data Element |

Description (Line E):

'E' is the access line used to describe an Element.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'E' | |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 3 | 6 | | Element code |
| 9 | 3 | | Line number |
| 12 | 1 | | Line type |
| 13 | 1 | | Skip or action type |
| | | | Numeric |
| 14 | 13 | | Data Element value |
| 27 | 54 | | Data Element value - Meaning |

Model Objects

Definition (Line K1):

The access line used to define a model entity, model relation or model F.I.C. is 'K1'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K1' | |
| 4 | 6 | | Object code |
| 10 | 36 | | Name of the object |
| 46 | 1 | | Type of the object |
| | | 'O' | Object, |
| | | 'R' | Relationship, |
| | | 'C' | Functional Integrity Constraint (F.I.C.). |
| 47 | 9 | | Number of instances |
| | | | Numeric |
| 56 | 6 | | Code of the implied Relation |
| | | | This field is used for the definition of an F.I.C. |
| 62 | 6 | | Parent object code |
| 68 | 10 | | Object comment |

Call of Properties in Object or Relat. (Line K3):

The line code used to call properties in an entity or a Model Relation is 'K3'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K3' | |
| 4 | 6 | | Object code |
| 10 | 3 | | Line number |
| 13 | 6 | | Element code |
| 19 | 1 | | Identifier in Segment |
| 20 | 3 | | Occurrences (COBOL "OCCURS" clause) |
| | | | Numeric |
| 23 | 2 | | Number of Data Elements in a group |

Model Relations

Definition (Line K1):

The access line used to define a model entity, model relation or model F.I.C. is 'K1'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K1' | |
| 4 | 6 | | Object code |
| 10 | 36 | | Name of the object |
| 46 | 1 | | Type of the object |
| | | 'O' | Object, |
| | | 'R' | Relationship, |
| | | 'C' | Functional Integrity Constraint (F.I.C.). |
| 47 | 9 | | Number of instances |
| | | | Numeric |
| 56 | 6 | | Code of the implied Relation |
| | | | This field is used for the definition of an F.I.C. |
| 62 | 6 | | Parent object code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 68 | 10 | | Object comment |

Call of Objects in Relation or F.I.C (Line K2):

The access line code used to call entities in a Relation or a F.I.C. is 'K2'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K2' | |
| 4 | 6 | | Model Relation code |
| 10 | 3 | | Line number |
| 13 | 6 | | Object code |
| 19 | 7 | | Occurrence ranking (minimal) |
| 26 | 7 | | Occurrence ranking (maximal) |
| 33 | 7 | | Average occurrence ranking |

Call of Properties in Object or Relat. (Line K3):

The line code used to call properties in an entity or a Model Relation is 'K3'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K3' | |
| 4 | 6 | | Object code |
| 10 | 3 | | Line number |
| 13 | 6 | | Element code |
| 19 | 1 | | Identifier in Segment |
| 20 | 3 | | Occurrences (COBOL "OCCURS" clause) |
| | | | Numeric |
| 23 | 2 | | Number of Data Elements in a group |

Model F.I.C.'s

Definition (Line K1):

The access line used to define a model entity, model relation or model F.I.C. is 'K1'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K1' | |
| 4 | 6 | | Object code |
| 10 | 36 | | Name of the object |
| 46 | 1 | | Type of the object |
| | | 'O' | Object, |
| | | 'R' | Relationship, |
| | | 'C' | Functional Integrity Constraint (F.I.C.). |
| 47 | 9 | | Number of instances |
| | | | Numeric |
| 56 | 6 | | Code of the implied Relation |
| | | | This field is used for the definition of an F.I.C. |
| 62 | 6 | | Parent object code |
| 68 | 10 | | Object comment |

Call of Objects in Relation or F.I.C (Line K2):

The access line code used to call entities in a Relation or a F.I.C. is 'K2'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'K2' | |
| 4 | 6 | | Model Relation code |
| 10 | 3 | | Line number |
| 13 | 6 | | Object code |
| 19 | 7 | | Occurrence ranking (minimal) |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 26 | 7 | | Occurrence ranking (maximal) |
| 33 | 7 | | Average occurrence ranking |

Data Structures

Definition (Line A):

'A' is the access line used to define a Data Structure.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'A' | |
| 3 | 2 | | Data Structure code |
| 5 | 30 | | Data Structure label |
| 35 | 44 | | Data Structure comment |
| 79 | 1 | | Type |
| 80 | 1 | | File reporting option |
| | | 'O' | file descriptions will include vet and update markers. This option is to be used only for files with vets, update markers, fields with variable repetitions, or with initial values. It is mandatory for generating error messages. |
| | | 'N' | File descriptions will not include vet and update markers. In this case, field lengths and addresses in the record will be indicated (default option) |
| | | 'E' | file descriptions will be presented in their input format with addresses , lengths, and initial values of the fields in the record |
| | | 'I' | file descriptions will be presented in internal format with addresses, lengths, and initial values of the fields in the record |

Segments

Definition (Line 2):

'2' is the access line used to define a Segment.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | '2' | |
| 4 | 4 | | Segment code |
| 8 | 1 | | Type of Segment definition line |
| | | 'L' | |
| 12 | 10 | | Structure Code value/Data Element |
| 22 | 6 | | Code of action code element |
| | | | In the BATCH SYSTEMS DEVELOPMENT FUNCTION: |
| | | | Enter the DATA ELEMENT CODE for the element used to identify the transaction type. The System will generate validation logic appropriate for creation, modification, deletion and implicit action codes, as well as user-defined transaction types. Six values are associated with this code. Validation and updates are automatic for these six values: |
| | | | . transaction 1 creation, . transaction 2 modification, . transaction 3 deletion, . transaction 4 modification . transaction 5 modification, . transaction 6 modification. |
| | | | If there is no ACTION CODE ELEMENT, this field remains blank, and the transaction type is a modification. In this case, presence specifications for the segment are entered in the MOD-4 : ACTN CODE VALUE / SEG PRES. field, and for the elements, in the MOD-4 field on the Call of Elements (-CE) screen. |
| | | | The CODE OF ACTION CODE ELEMENT and the values must be entered on only one segment of the data structure, preferably on the common part '00'. |
| 28 | 5 | | Create: Actn code value / Seg pres. |
| | | | (Specific to the Batch Systems Development function). |
| | | | ACTION CODE VALUE: |
| | | | On the '00' segment, enter the value that stands for "create" for this file: Example: 'ADD'. Note: for alphabetic characters use quotes. |
| | | | SEGMENT PRESENCE: |
| | | | On the non-00 segments, enter the presence specifications for the individual segment. |
| | | 'O' | Obligatory: the segment must be present on a "create" |
| | | 'I' | Invalid: the segment must not be present on a "create" |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | 'F' | Optional (default). |
| 33 | 5 | | Modify: action code value/ Seg pres. |
| | | | (Specific to the Batch Systems Development function). |
| | | | ACTION CODE VALUE: |
| | | | On the '00' segment, enter the value that stands for "modify" for this file: Example: 'CHG'. Note: for alphabetic characters use quotes. |
| | | | SEGMENT PRESENCE: |
| | | | On the non-00 segments, enter the presence specifications for the individual segment. |
| | | 'O' | Obligatory: the segment must be present on a "modify" |
| | | 'I' | Invalid: the segment must not be present on a "modify" |
| | | 'F' | Optional (default) |
| 38 | 5 | | Delete: actn code value / Seg pres. |
| | | | (Specific to the Batch Systems Development function). |
| | | | ACTION CODE VALUE: |
| | | | On the '00' segment, enter the value that stands for "delete" for this file: Example: 'DEL'. Note: for alphabetic characters use quotes. |
| | | | SEGMENT PRESENCE: |
| | | | On the non-00 segments, enter the presence specifications for the individual segment. |
| | | 'O' | Obligatory: the segment must be present on a "delete" |
| | | 'I' | Invalid: the segment must not be present on a "delete" |
| | | 'F' | Optional (default). |
| 43 | 5 | | Mod-4: actn code value / Seg pres. |
| | | | (Specific to the Batch Systems Development function). |
| | | | ACTION CODE VALUE: |
| | | | On the '00' segment, enter the value that stands for implicit action codes - (creates or modifications). Note: for alphabetic characters use quotes. |
| | | | SEGMENT PRESENCE: |
| | | | On the non-00 segments, enter the presence specifications for the individual segment. |
| | | 'O' | Obligatory: the segment must be present. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | 'I' | Invalid: the segment must not be present. |
| | | 'F' | Optional (default). |
| 48 | 5 | | Mod-5: actn code value / Seg pres. |
| | | | (Specific to the Batch Systems Development function). |
| | | | ACTION CODE VALUE: |
| | | | On the '00' segment, enter the value that stands for this user-defined action. Note: for alphabetic characters use quotes. |
| | | | SEGMENT PRESENCE: |
| | | | On the non-00 segments, enter the presence specifications for the individual segment. |
| | | 'O' | Obligatory: the segment must be present. |
| | | 'I' | Invalid: the segment must not be present. |
| | | 'F' | Optional (default). |
| 53 | 5 | | Mod-6: actn code value / Seg pres. |
| | | | (Specific to the Batch Systems Development function). |
| | | | ACTION CODE VALUE: |
| | | | On the '00' segment, enter the value that stands for this user-defined action. Note: for alphabetic characters use quotes. |
| | | | SEGMENT PRESENCE: |
| | | | On the non-00 segments, enter the presence specifications for the individual segment. |
| | | 'O' | Obligatory: the segment must be present. |
| | | 'I' | Invalid: the segment must not be present. |
| | | 'F' | Optional (default) |
| 58 | 1 | | Create: segment presence |
| 59 | 1 | | Modify: segment presence |
| 60 | 1 | | Delete: segment presence |
| 61 | 1 | | Mod-4 : segment presence |
| 62 | 1 | | Mod-5 : segment presence |
| 63 | 1 | | Mod-6 : segment presence |
| 64 | 4 | | Occurs in Table |
| 68 | 9 | | Estimated number of instances |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | Numeric |
| 77 | 1 | | Continuation line indicator |

Description (Line 3):

'3' is the access line used to call Elements into a Segment.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '3' | |
| 3 | 4 | | Segment code |
| 7 | 3 | | Line number |
| 10 | 6 | | Element code |
| 16 | 18 | | Element short name |
| 34 | 10 | | Data Element internal format |
| 44 | 1 | | Element internal use |
| 45 | 3 | | Occurrences (COBOL "OCCURS" clause) |
| | | | Numeric |
| 48 | 2 | | Number of Data Elements in a group |
| 50 | 1 | | Identifier in Segment |
| 51 | 1 | | Creation |
| 52 | 1 | | Modification |
| 53 | 1 | | Deletion |
| 54 | 1 | | Type 4 |
| 55 | 1 | | Type 5 |
| 56 | 1 | | Type 6 |
| 57 | 1 | | Class (alpha/numeric) |
| 58 | 1 | | Operators (and/or) |
| 59 | 1 | | Negation (NOT) |
| | | 'N' | Negation ('NOT' is generated). |
| | | blank | No negation. |
| 60 | 1 | | Type: validation, update, values |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 61 | 10 | | Values / sub-function code |
| 71 | 2 | | Update target / first part |
| 73 | 2 | | Update target / second part |
| 75 | 6 | | Update target / last part |

Pactables Sub-Schemas and Sub-Systems (Line 21):

The line code used to define all sub-schemas and sub-systems of a Table is '21'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | '21' | |
| 4 | 4 | | Segment code |
| 8 | 1 | | Type of Segment definition line |
| | | 'S' | Sub-schema definition. |
| | | 'Y' | Sub-system definition. |
| 9 | 1 | | Sub-schema / sub-system number |
| 10 | 30 | | Sub-schema/sub-system name |
| 40 | 4 | | Occurs in Table |

Reports

Definition (Line B):

'B' is the line code used to define a Report.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'B' | |
| 3 | 3 | | Report Code |
| 6 | 30 | | Report name |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 36 | 36 | | Comments |
| 72 | 1 | | Nature code |
| 73 | 1 | | Type |
| 74 | 3 | | Line length (maximum) |
| | | | Numeric |
| 77 | 2 | | No. of digits left of the decimal |
| | | | Numeric |
| 79 | 2 | | No. of digits right of the decimal |
| | | | Numeric |

Report Layout Description (Line 4):

'4' is the line code used to describe a Report layout.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '4' | |
| 3 | 3 | | Report Code |
| 6 | 2 | | Line number |
| 8 | 2 | | Constant part number |
| | | | Numeric |
| 10 | 1 | | Number of printed literals part |
| 11 | 1 | | Line skip/page break |
| | | | Numeric |
| 12 | 1 | | Char. set option: special printer |
| 15 | 66 | | Edition label |

Report Characteristics Description (Lines 5, E):

Batch Form '5' (type E) is used to describe the report characteristics.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '5' | |
| 3 | 3 | | Report Code |
| 14 | 1 | | Line type |
| | | 'E' | Default value |
| 15 | 3 | | Length of the variable part |
| 24 | 1 | | Option |
| | | 'Blank' | Print options are generated according to the hardware variant indicated at the library level. |
| | | | In the case of conversion libraries, the print options are automatically reformulated according to the library variant. |
| | | 'N' | Prohibits any automatic reformulation of the print option, in a conversion library. |
| | | '*' | Generation of 'WRITE BEFORE' statement. |
| 25 | 2 | | Lines per page |
| | | | PURE NUMERIC FIELD |
| | | | Default option: 60. |
| 27 | 4 | | No. of instances in category table |
| | | | PURE NUMERIC FIELD |
| | | | Enter the number of positions to allocate to store the different categories in the report (upon generation). |
| | | '100' | Default. |
| | | '0000' | Rather than using the category table to control the organization of printing the categories, the categories are printed directly. |
| | | | Note: If the number of positions is higher than 1000, the table is not generated. |
| 31 | 2 | | Section priority |
| | | | This field is used with hardware requiring program segmentation due to small memory capacity. For information, consult a COBOL manual. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | Generates a segment type overlay between print functions in a program. It should only be used if input data structures to print programs are sorted by report code and if the COBOL variant is ANSI. Priorities less than 50 generate an overlay only in association with the 'SEGMENT LIMIT' clause, to be inserted in the ENVIRONMENT DIVISION. |
| 33 | 13 | | Comments |
| | | | The comment entered on the screen top refers to the whole report. Comments entered on the screen body normally refer to the individual lines. |
| 46 | 35 | | Condition of execution |
| | | | The conditions indicated in the screen top are relevant to the execution of the Report. |
| | | | The conditions indicated in the screen body are relevant to the execution of the Category of the Report. |
| | | | Input format: |
| | | | Use the COBOL format to enter conditions but do not enter 'IF', nor GO TO, and do not enter any period. |

List of Categories (Line 5):

'5' is the line code used to describe the report categories.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '5' | |
| 3 | 3 | | Report Code |
| 6 | 2 | | Category |
| | | | (maximum of 39 lines per category.) |
| | | 'AB to ZY' | The value entered here is used to differentiate categories from one another. Report lines are grouped together according to the conditions under which they will be printed (totaled, etc...). |
| | | | Leaving gaps in the category sequence will facilitate future modifications. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | Categories containing a detail line with elements to be totaled - (TYPE OF LINE = '*' or 'T'): |
| | | | .can only contain one detail line, |
| | | | .cannot contain a total line, |
| | | | .cannot be repetitive, |
| | | | .can contain other ordinary lines. |
| | | | Categories used for the lines containing the totals - (TYPE OF LINE = '0' to '9'): |
| | | | .can contain several total lines, |
| | | | .cannot have a detail line, |
| | | | .cannot be repetitive, |
| | | | .can contain other ordinary lines. |
| | | 'ZZ' | Prohibited. |
| | | 'AA' | Not recommended. |
| 8 | 3 | | Line number |
| 14 | 1 | | Type of line |
| | | | This field is used to identify the type of category. |
| | | | To designate a Header, repetitive area, or Footer: |
| | | 'A' | This value applies to repetitive categories only. This indicates the first line of a top-of-page category (header). Headers are automatically printed at the top of each page of a report. They are also printed when the repetitive category lines exceed the number of lines per page allowed for the report, causing a new page to be printed. |
| | | 'E' | In Batch forms, line which describes the Report (general characteristics and condition). |
| | | 'T' | Indicates the first line of a category printed several times (repetitive category). This value causes the generation of a subscript which controls the number of repetitions. This number may be fixed or variable. |
| | | | For a fixed number: |
| | | | .enter a number in the TOTALING LINE INDICATOR field |
| | | | For a variable number: |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | .enter a three-character code in the TOTALING LINE INDICATOR field. (The code was defined on the Work Areas (-W) screen for use as the subscript field. Procedural code is used to move in the values.) OR .use the standard PACBASE index (Jddrcc), generated for the category: Note: ddr = REPORT CODE, cc = CATEGORY OF REPORT (repetitive) See SOURCE FIELD - LAST PART on the Report Call of Elements (-CE) screen, with value '*cc'. |
| | | 'Z' | This value applies to repetitive categories only. This indicates the first line of an end-of-page category (footer). Footers are automatically printed when the repetitive category lines exceed the number of lines per page allowed for that report. |
| | | | To identify detail lines with fields to accumulate: |
| | | '*' | This indicates a detail line containing fields whose values are to be accumulated for totaling. The lines will be printed in the report. Note: The data elements to total are identified on the Report Call of Elements screen by entering 'T' in OPERATION ON SOURCE FIELD. All elements are conditioned by report category. (See Subchapter "CALL OF DATA ELEMENTS (-CE)".) |
| | | | A category containing a detail line: . can contain only one detail line, . cannot contain a total line, . cannot be iterative, . can include other ordinary lines. |
| | | | The logic for data elements to be totaled is generated only if the conditions specified for the '*' line category are met. |
| | | 'T' | Same as '*', but the category containing this line is not to be printed. |
| | | | Note: For information concerning other lines that may or may not be included with lines of this type, see CATEGORY OF REPORT. |
| | | | One program may use several reports. There can only be 12 '*' and 'T' type lines (combined) per program. |
| | | | To identify lines displaying accumulated totals: |
| | | '0' | Indicates a line for Grand Totals. Note: Grand Totals may only be requested if there is at least one Total at a control break level. At least one control break has to be specified for a file on the -CD screen. |
| | | '1 to 9' | Indicates a line for totaling at the control break level corresponding to this value. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | A category containing a total line: . may contain several of them, . cannot contain a detail line, . cannot be iterative, . can include other ordinary lines. |
| | | | See CATEGORY OF REPORT for information on other lines that may or may not be included in a category with totaling-type lines. |
| | | | NOTE: A detail line may be defined in a different report. For example, a summary report based on accumulations from other reports may be needed. This can be done using the following technique: The STRUCTURE NUMBER assigned to the detail line of the other report is not used on the summary report's Call of Elements screen, and on its Description (-D) screen, the TYPE OF LINE value is entered and the TOTALING LINE INDICATOR will be comprised of the LAST CHARACTER OF REPORT CODE of the report containing the detail line, followed by its STRUCTURE NUMBER. Only the totaled data elements will be printed, at the designated control break level. |
| 15 | 3 | | Characteristics |
| | | | On a line that has fields being totaled (TYPE OF LINE values '0' to '9'), which has a detail line described in a different report, enter the following: |
| | | | .first character: LAST CHARACTER OF REPORT CODE of the report containing the description, |
| | | | .2nd and 3rd characters: STRUCTURE NUMBER. |
| | | | On the first line of a repetitive category (TYPE OF LINE = 'T'), this value causes the generation of a subscript which controls the number of repetitions. This number may be fixed or variable. |
| | | | For a fixed number: |
| | | | .enter an absolute number value. |
| | | | For a variable number: |
| | | 'blank' | .enter the three character code defined on the Work Areas (-W) screen for use as the subscript field. (The values are determined via Procedural Code.) OR .use the standard PACBASE index (Jddrcc), generated for the category. |
| 18 | 2 | | Structure number |
| | | | Numeric |
| 20 | 2 | | Constant part number |
| | | | Numeric |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 22 | 2 | | Line skip/page break |
| | | | Numeric |
| 24 | 1 | | Line skip type |
| 27 | 2 | | Function code |
| 29 | 2 | | Sub-function code |
| 33 | 13 | | Comments |
| | | | The comment entered on the screen top refers to the whole report. Comments entered on the screen body normally refer to the individual lines. |
| 46 | 35 | | Condition of execution |
| | | | The conditions indicated in the screen top are relevant to the execution of the Report. |
| | | | The conditions indicated in the screen body are relevant to the execution of the Category of the Report. |
| | | | Input format: |
| | | | Use the COBOL format to enter conditions but do not enter 'IF', nor GO TO, and do not enter any period. |

Description of Structures (Line 6):

'6' is the line code used to call Elements into Structures.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '6' | |
| 3 | 3 | | Report Code |
| 6 | 2 | | Structure number |
| | | | Numeric |
| 8 | 3 | | Starting address (column number) |
| | | | Numeric |
| 11 | 1 | | Data element line number |
| 12 | 6 | | Element code |
| 18 | 2 | | Continuation of D.S. Description |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | '**' | Enter '**' to specify a continuation line, in which you indicate the continuation of a condition. |
| 20 | 14 | | Output Format |
| | | | (Default option: INTERNAL FORMAT) |
| | | | This is the format of a data element as it is used in a printed report, or in a screen as a display field. It is automatically transferred in the description of printed reports, screens and segments. |
| | | | It must be coded like a COBOL picture. USAGE is always DISPLAY. |
| | | | In previous versions, this field was used to generate the BLANK WHEN ZERO clause, which may be displayed in this field. |
| | | | When creating or updating a data element, the BLANK WHEN ZERO CLAUSE field must be used for this purpose. |
| | | | For data elements representing a date, it is possible to assign a symbolic format: |
| | | | Display type formats (input): |
| | | 'D' | Without century (picture (x6)). |
| | | 'C' | With century (picture (x8)). |
| | | | Internal type formats: |
| | | 'I' | Without century (picture x(6)). |
| | | 'S' | With century (picture x(8)). |
| | | | Extended type formats (output) (with slashes): |
| | | 'E' | With century (picture x(8)). |
| | | 'M' | With century (picture x(10)). |
| | | 'G' | Gregorian format (picture x(10)). |
| | | 'T' | TIME format. |
| | | 'TS' | TIMESTAMP format |
| | | | PACMODEL function: This field may be omitted for a property. |
| | | | For details on the use of the formats with the various types of Database Blocks, see the summary tables in chapter "COLUMNS: DATA ELEMENTS" of the "Relational SQL Database Description" Reference Manual. |
| 34 | 1 | | Operation on source field |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 35 | 1 | | Working-Storage Prefix of Source |
| 36 | 2 | | Source field - first part |
| 38 | 2 | | Source field - second part |
| 40 | 6 | | Code of source field |
| 46 | 3 | | Source field - last part |
| 49 | 32 | | Execution condition |

On-Line Screens

Definition (Line H):

'H' is the line code used to define a Dialogue or a Screen information (name, number of lines and columns, etc.), and a second part, which contains:

- With a blank in the continuation field: the attributes, documentation call fields (PFkeys or characters), initialization character for entry fields;
- With '*' in the continuation field: the external name of the program, the external name of the map, the transaction code.

Usually, only one 'H' line code with the attributes is necessary to define a dialogue and only one 'H' line code with the external names is necessary to define a screen: in general, a screen takes on the attributes defined at the dialogue level.

However, both layout formats of line code 'H' can be entered to define a Dialogue or a Screen.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'H ' | |
| 4 | 2 | | Dialogue code |
| 6 | 4 | | Screen code within the dialogue |
| 10 | 30 | | Dialogue or screen name |
| 40 | 2 | NUMER. | Screen size - number of lines |
| | | | Numeric |
| 42 | 3 | NUMER. | Screen size - number of columns |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | | Numeric |
| 45 | 1 | | Label type |
| 46 | 2 | NUMER. | Number of tabs per line |
| | | | Numeric |
| 48 | 2 | | Transactional language variant |
| 50 | 1 | | Optional Command Lines Set/BEFORE |
| 51 | 1 | | Control cards in front of map |
| 52 | 1 | | Optional Command Lines Set/AFTER |
| 53 | 1 | | Control cards in back of map |
| 54 | 1 | | Intensity attribute - label |
| 55 | 1 | | Intensity attribute - display field |
| 56 | 1 | | Intensity attribute - input field |
| 57 | 1 | | Intensity attribute - error message |
| 58 | 1 | | Intensity attribute-erroneous field |
| 59 | 1 | | Color attribute - label |
| 60 | 1 | | Color attribute - display field |
| 61 | 1 | | Color attribute - input field |
| 62 | 1 | | Color attribute - error message |
| 63 | 1 | | Color attribute - erroneous field |
| 64 | 1 | | Presentation attribute - label |
| 65 | 1 | | Presentation attribute-display field |
| 66 | 1 | | Presentation attribute - input field |
| 67 | 1 | | Presentation attribute-error message |
| 68 | 1 | | Presentation att. - erroneous field |
| 70 | 2 | | Help character: screen help |
| 72 | 2 | | Help character: data element help |
| 74 | 1 | | Initialization character: variables |
| 78 | 2 | | Screen type |
| 80 | 1 | | Continuation line |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 2 | 2 | | Line code |
| | | 'H ' | |
| 4 | 2 | | Dialogue code |
| 6 | 4 | | Screen code within the dialogue |
| 10 | 30 | | Dialogue or screen name |
| 40 | 2 | NUMER. | Screen size - number of lines |
| | | | Numeric |
| 42 | 3 | NUMER. | Screen size - number of columns |
| | | | Numeric |
| 45 | 1 | | Label type |
| 46 | 2 | NUMER. | Number of tabs per line |
| | | | Numeric |
| 48 | 2 | | Transactional language variant |
| 50 | 1 | | Optional Command Lines Set/BEFORE |
| 51 | 1 | | Control cards in front of map |
| 52 | 1 | | Optional Command Lines Set/AFTER |
| 53 | 1 | | Control cards in back of map |
| 54 | 8 | | External name of program |
| 62 | 8 | | External name of map |
| 70 | 8 | | Transaction code |
| 78 | 2 | | Screen type |
| 80 | 1 | | Continuation line |

Dialog Complement (Line H3):

'H3' is the line code used to enter the Dialogue Complement. It must be preceded by line code 'H', which specifies the Dialogue Code.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'H3' | |
| 4 | 2 | | Common area - data structure code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 6 | 1 | | Organization |
| 7 | 8 | | External name of error message file |
| 15 | 4 | | First screen code of the dialogue |
| 19 | 6 | | Database Block code |
| 25 | 4 | NUMER. | Complementary common area length |
| | | | Numeric |
| 29 | 47 | | Options |

Description (Line I):

'I' is the line code used to describe a screen.

Note :

It must be preceded by a line code H which specifies the dialogue Code.

On the lines codes of screens description (I-type line code), enter the ? character in the column 31 to blank out the 'label type' field.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'I' | |
| 3 | 3 | | Line number |
| 6 | 6 | | Element code |
| 12 | 1 | | Positioning type |
| 13 | 2 | NUMER. | Line number positioning |
| | | | Numeric |
| 15 | 3 | NUMER. | Column number positioning |
| | | | Numeric |
| 18 | 1 | | Nature of the data element |
| 19 | 1 | | Label type |
| 20 | 1 | | Intensity attribute - label |
| 21 | 1 | | Intensity attribute - data |
| 22 | 1 | | Presentation attribute - label |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 23 | 1 | | Presentation attribute - data |
| 24 | 1 | | Color attribute - label |
| 25 | 1 | | Color attribute - data |
| 26 | 1 | | Cursor default position/skip option |
| 27 | 2 | NUMER. | Horizontal repetitions number |
| 29 | 2 | NUMER. | Vertical repetitions number |
| 31 | 1 | | Presence validation of data element |
| 32 | 1 | | Validation conditions/set variables |
| 33 | 1 | | Update option |
| 34 | 4 | | Update target: segment code |
| 38 | 6 | | Update target / last part |
| 44 | 1 | | Working-Storage Prefix of Source |
| 45 | 4 | | Source segment code |
| 49 | 6 | | Code of source field |
| 60 | 2 | NUMER. | Level |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | T | |
| 3 | 3 | | Line number |
| 6 | 6 | | Element code |
| 12 | 1 | | Positioning type |
| 13 | 2 | NUMER. | Line number positioning |
| | | | Numeric |
| 15 | 3 | NUMER. | Column number positioning |
| | | | Numeric |
| 18 | 1 | | Nature of the data element |
| 19 | 1 | | Label type |
| 20 | 1 | | Intensity attribute - label |
| 21 | 1 | | Intensity attribute - data |
| 22 | 1 | | Presentation attribute - label |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 23 | 1 | | Presentation attribute - data |
| 24 | 1 | | Color attribute - label |
| 25 | 1 | | Color attribute - data |
| 26 | 1 | | Cursor default position/skip option |
| 27 | 2 | NUMER. | Horizontal repetitions number |
| 29 | 2 | NUMER. | Vertical repetitions number |
| 31 | 1 | | Type of literal |
| | | | Defines the contents of the next field, which is displayed on the Call of Elements (-CE) with OPERATION CODE 2. |
| | | blank | The field contains a fixed label value. |
| | | 'T' | The field contains an initial value automatically displayed when the Screen is invoked. |
| | | 'P' | The field contains a presentation value used for the Screen simulation only. |
| | | 'A' | This value indicates that the following label is made up of one character repeated more than 30 times. |
| | | | INPUT EXAMPLE: |
| | | | LABEL |
| | | | T LITERALS |
| | | | A 045- |
| | | | The corresponding label is a line of 45 dashes. |
| | | | IBM 36, IBM 38, IBM AS/ 400: |
| | | 'Y' | This value specifies that the next field contains an INDICATOR number for attribute positioning. |
| 32 | 30 | | Displayed literal |

Call of Segments (Line H2):

'H2' is the line code used to call segments into a screen.

It must be preceded by a line code 'H' which specifies the Screen Code.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| | | 'H2' | |
| 4 | 1 | | Category indicator (screen) |
| 5 | 4 | | Segment code in program |
| | | | This group column contains the following elementary columns: |
| | | | CDSTPG (Code of Data Structure in Program), CRECPG (Code of Record in Program) |
| 9 | 2 | | Line number |
| 11 | 1 | | Access mode |
| | | 'S' | Sequential (default option). |
| | | 'R' | Random - Direct (indexed sequential organization only). |
| | | 'D' | Dynamic (VSAM files only - 'V' organization) |
| 12 | 1 | | Use in reception |
| 13 | 1 | | Use in display |
| 14 | 4 | | Preceding segment code |
| 18 | 14 | | Access key source |
| | | | This group column contains the following elementary columns: |
| | | | CSEGSR (Code of Source Segment), CDELSR (Code of Source Data Element). |
| 32 | 6 | | Element code |
| 38 | 1 | | Control break indicator for display |
| 39 | 1 | | Organization |
| 40 | 1 | | Generated description type |
| 41 | 8 | | External name of the file |
| 49 | 2 | | Data Structure code |
| 51 | 2 | | Code |
| 53 | 1 | | Sub-schema / sub-system number |
| 54 | 2 | NUMER. | Level |

Call of Macro-Structures (Line M):

Macro-structures are called using the line code 'M'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'M' | |
| 3 | 2 | | Line number |
| 5 | 1 | | Expansion option for Macro-Struct. |
| | | blank | The macro-structure lines are expanded in the calling programs during the update |
| | | 'N' | No expansion of macro-structure lines during the update |
| 6 | 1 | | Delimiter of parameter values |
| 7 | 6 | | Macro-structure code |
| 13 | 50 | | Parameter identifier |
| 80 | 1 | | Continuation line |

Program Beginning Insertions (Line D):

The 'Beginning of Program' is modified using the line code 'D'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'D' | |
| 4 | 2 | | Section to generate |
| 7 | 2 | | Paragraph title |
| 9 | 3 | | Line number |
| 13 | 66 | | Instruction |

Working Areas (Line 7):

The Work areas and Linkage areas are described using the line code '7'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '7' | |
| 5 | 2 | | Line beginning |
| 7 | 3 | | Line number |
| 10 | 1 | | Type of line or Data element format |
| | | | Type of line values: |
| | | blank | Data entered in the LEVEL AND SECTION and WORK AREA DESCRIPTION fields are to be generated as entered. |
| | | '.' | Continuation character for a literal. |
| | | '*' | Comment. Data entered in the LEVEL AND SECTION and WORK AREA DESCRIPTION fields contain comments to be inserted into the generated Program (Table size is not considered as comment, see line's last field). |
| | | '\$' | This value appears in column 7 of the generated COBOL and the other Elements of the WORKING line appear as it is. |
| | | 'A' | Call of an eBusiness Application. This call is fully documented in the 'COBOL API User's Guide'. |
| | | 'F' | Call of a Data Structure. |
| | | | When 'F' is entered, the system responds with a formatted line which is used to facilitate data entry. The fields are the same as those used on the Call of Data Structures (-CD) screen for D.S. with ORGANIZATION = 'W' or 'L'. |
| | | | .Data Structure code in the Program. |
| | | | .Data Structure code in the Library. |
| | | | .Segment selection (enter the Segment code without an asterisk). |
| | | | (A segment code can only be renamed in batch). |
| | | | .Non-Printing Data Structure format (1 to 8). |
| | | | .Record type / Use within D.S. (I, E or S). |
| | | | .Level number (COBOL) of the record (1 to 5). |
| | | | .Organization. |
| | | | .Sub-schema number. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | Type 'F' '-W' lines are processed as Data Structure call lines (-CD) only for batch. |
| | | | If two Type 'F' '-W' lines referring to the same Data Structure (same Data Structure code in the Program) are separated, they will nevertheless be generated one after the other. |
| | | | Element format values: |
| | | 'E' | Use of the Input format of a Data Element. |
| | | 'I' | Use of the Internal format of a Data Element. |
| | | 'S' | Use of the Output format of a Data Element. |
| | | | For these format types, the presence of the Data Element in the Specifications Dictionary is checked. A cross-reference is established, which prohibits the deletion of the Data Element whenever the lines in which it is called have not been deleted themselves. |
| | | | If the Element does not exist in the Specifications Dictionary, the System sends a warning. |
| | | | When a global replacement is required (.C2), the Element is not checked but the cross-references will still be created. |
| | | | For these three format types, the entered data-name must therefore have the following format: |
| | | | W-DDSS-EEEEEE where: |
| | | | W = a working-storage prefix, |
| | | | DDSS = a given Data Structure and Segment code, |
| | | | EEEEEE = a Data Element code which exists in the Specifications Dictionary. |
| | | | The corresponding format is automatically attributed by the System. |
| | | | For IMS sub-monitors: |
| | | 'M' | Sub-monitor; enter the code of the sub-monitor in the LEVEL OR SECTION field. |
| | | 'C' | Call of a screen into the sub-monitor named above. |
| | | | Enter the SCREEN CODE of the screen belonging to the sub-monitor in the LEVEL OR SECTION field, followed by a space and a 'D' for Dynamic call or 'S' for Static. |
| | | | Example: C OOSCRN D |
| | | | Note: Enter one SCREEN CODE per 'C'-type line. |
| 11 | 17 | | Level or section |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 28 | 48 | | Work data declaration |
| 76 | 5 | | Table size (occurs clause) |

Procedural Code (Line P):

Procedural code is written using the line code 'P'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'P' | |
| 3 | 2 | | Function code |
| | | 'AA-99' | |
| | | '\$n' | In a macro-structure, the function code can be parameterized. |
| 5 | 2 | | Sub-function |
| | | 'AA-99' | Sub-function code. |
| | | '\$n' | In a macro-structure, the sub-function code can be parameterized. |
| 7 | 3 | | Line number |
| 10 | 3 | | Operator |
| 13 | 32 | | Operand |
| 45 | 2 | NUMER. | Level |
| 47 | 2 | | Condition Type |
| 49 | 32 | | Execution condition |

Programs

Definition (Line 0):

'0' (zero) is the line code used to define a program.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '0' | Zero |
| 7 | 6 | | Program code |
| 13 | 6 | | Code for sequence of generation |
| 19 | 30 | | Program name |
| 49 | 1 | | Type of COBOL |
| 50 | 1 | | Order of insertion in COBOL Library |
| 51 | 1 | | COBOL numbering and alignment option |
| 54 | 1 | | SQL indicators generation with '-' |
| 55 | 1 | | Optional Command Lines Set/BEFORE |
| 56 | 1 | | Optional Command Lines Set/AFTER |
| 57 | 8 | | COBOL program id |
| 65 | 1 | | Programming mode |
| 66 | 1 | | Type and structure of program |
| 67 | 1 | | Type of presence validation |
| 68 | 1 | | Program classification code |

Call of Data Structures (Line 1):

'1' is the line code used for the 'Call of Data Structures'.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '1' | |
| 3 | 2 | | Data structure code in the program |
| 5 | 2 | | Data Structure code |
| 7 | 6 | | External name |
| 13 | 1 | | Organization |
| 14 | 1 | | Access mode |
| 15 | 1 | | Recording mode |
| 16 | 1 | | Opening mode |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 17 | 1 | | Unit type |
| 18 | 5 | | Block size |
| | | | Numeric |
| 23 | 1 | | Block size unit type |
| 24 | 10 | | File status |
| 34 | 6 | | Indexed Data Structure Access Key |
| | | | Required for indexed Data Structures: Enter the DATA ELEMENT CODE of the access key Element. |
| 40 | 1 | NUMER. | Number of control breaks |
| | | | Numeric |
| 41 | 1 | | File matching level number |
| | | | Numeric |
| 42 | 1 | | Usage |
| 43 | 6 | | Element code |
| 49 | 2 | | Resulting file data structure code |
| 51 | 2 | | Source or error data structure code |
| 53 | 1 | | Transaction control break level |
| 59 | 4 | | Physical Unit Type |
| 63 | 1 | | Unit Complement |
| 64 | 9 | | Sort key / seg select / report codes |
| 73 | 1 | | Format type |
| 74 | 1 | | Selected description |
| 75 | 1 | | Generated description type |
| 76 | 1 | | Level |
| 77 | 2 | | Line beginning |
| 79 | 2 | | Continuation of D.S. description |

Call of Macro-Structures (Line M):

Macro-structures are called using the line code 'M'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'M' | |
| 3 | 2 | | Line number |
| 5 | 1 | | Expansion option for Macro-Struct. |
| | | blank | The macro-structure lines are expanded in the calling programs during the update |
| | | 'N' | No expansion of macro-structure lines during the update |
| 6 | 1 | | Delimiter of parameter values |
| 7 | 6 | | Macro-structure code |
| 13 | 50 | | Parameter identifier |
| 80 | 1 | | Continuation line |

Program Beginning Insertions (Line D):

The 'Beginning of Program' is modified using the line code 'D'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'D' | |
| 4 | 2 | | Section to generate |
| 7 | 2 | | Paragraph title |
| 9 | 3 | | Line number |
| 13 | 66 | | Instruction |

Working Areas (Line 7):

The working and linkage areas are described using the line code '7'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '7' | |
| 5 | 2 | | Line beginning |
| 7 | 3 | | Line number |
| 10 | 1 | | Type of line or Data element format |
| | | | Type of line values: |
| | | blank | Data entered in the LEVEL AND SECTION and WORK AREA DESCRIPTION fields are to be generated as entered. |
| | | '.' | Continuation character for a literal. |
| | | '*' | Comment. Data entered in the LEVEL AND SECTION and WORK AREA DESCRIPTION fields contain comments to be inserted into the generated Program (Table size is not considered as comment, see line's last field). |
| | | '\$' | This value appears in column 7 of the generated COBOL and the other Elements of the WORKING line appear as it is. |
| | | 'A' | Call of an eBusiness Application. This call is fully documented in the 'COBOL API User's Guide'. |
| | | 'F' | Call of a Data Structure. |
| | | | When 'F' is entered, the system responds with a formatted line which is used to facilitate data entry. The fields are the same as those used on the Call of Data Structures (-CD) screen for D.S. with ORGANIZATION = 'W' or 'L'. |
| | | | .Data Structure code in the Program. |
| | | | .Data Structure code in the Library. |
| | | | .Segment selection (enter the Segment code without an asterisk). |
| | | | (A segment code can only be renamed in batch). |
| | | | .Non-Printing Data Structure format (1 to 8). |
| | | | .Record type / Use within D.S. (I, E or S). |
| | | | .Level number (COBOL) of the record (1 to 5). |
| | | | .Organization. |
| | | | .Sub-schema number. |
| | | | Type 'F' '-W' lines are processed as Data Structure call lines (-CD) only for batch. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | If two Type 'F' '-W' lines referring to the same Data Structure (same Data Structure code in the Program) are separated, they will nevertheless be generated one after the other. |
| | | | Element format values: |
| | | 'E' | Use of the Input format of a Data Element. |
| | | 'I' | Use of the Internal format of a Data Element. |
| | | 'S' | Use of the Output format of a Data Element. |
| | | | For these format types, the presence of the Data Element in the Specifications Dictionary is checked. A cross-reference is established, which prohibits the deletion of the Data Element whenever the lines in which it is called have not been deleted themselves. |
| | | | If the Element does not exist in the Specifications Dictionary, the System sends a warning. |
| | | | When a global replacement is required (.C2), the Element is not checked but the cross-references will still be created. |
| | | | For these three format types, the entered data-name must therefore have the following format: |
| | | | W-DDSS-EEEEEE where: |
| | | | W = a working-storage prefix, |
| | | | DDSS = a given Data Structure and Segment code, |
| | | | EEEEEE = a Data Element code which exists in the Specifications Dictionary. |
| | | | The corresponding format is automatically attributed by the System. |
| | | | For IMS sub-monitors: |
| | | 'M' | Sub-monitor; enter the code of the sub-monitor in the LEVEL OR SECTION field. |
| | | 'C' | Call of a screen into the sub-monitor named above. |
| | | | Enter the SCREEN CODE of the screen belonging to the sub-monitor in the LEVEL OR SECTION field, followed by a space and a 'D' for Dynamic call or 'S' for Static. |
| | | | Example: C OOSCRN D |
| | | | Note: Enter one SCREEN CODE per 'C'-type line. |
| 11 | 17 | | Level or section |
| 28 | 48 | | Work data declaration |
| 76 | 5 | | Table size (occurs clause) |

Procedural Code (Line P):

Procedural code is written using the line code 'P'.

Since it contains no program or screen code, this line must always be preceded by a program or screen definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'P' | |
| 3 | 2 | | Function code |
| | | 'AA-99' | |
| | | '\$n' | In a macro-structure, the function code can be parameterized. |
| 5 | 2 | | Sub-function |
| | | 'AA-99' | Sub-function code. |
| | | '\$n' | In a macro-structure, the sub-function code can be parameterized. |
| 7 | 3 | | Line number |
| 10 | 3 | | Operator |
| 13 | 32 | | Operand |
| 45 | 2 | NUMER. | Level |
| 47 | 2 | | Condition Type |
| 49 | 32 | | Execution condition |

COBOL Source Lines (Line FC):

Source Code is written using the line code 'FC'.

Since it contains no program code, this line must always be preceded by a program definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'FC' | |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 4 | 2 | | Function code |
| | | 'AA-99' | |
| | | '\$n' | In a macro-structure, the function code can be parameterized. |
| 6 | 2 | | Sub-function |
| | | 'AA-99' | Sub-function code. |
| | | '\$n' | In a macro-structure, the sub-function code can be parameterized. |
| 8 | 3 | | Line number |
| 11 | 3 | | Operator |
| 14 | 67 | | Source line |

Pure COBOL Source Lines (Line 9):

Pure COBOL Source Code (-9) lines may be entered on line code '9'.

Since it contains no program code, this line must always be preceded by a program definition line.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | '9' | |
| 3 | 6 | | COBOL line number |
| 9 | 1 | | Continuation line |
| 10 | 65 | | COBOL instruction |
| 75 | 6 | | End of COBOL line |

Database Blocks (Hierarchical)

Definition (Line L1):

'L1' is the line code used to define a Database Block.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 2 | 2 | | Line code |
| | | 'L1' | |
| 4 | 6 | | Database Block code |
| 10 | 36 | | Name of the block |
| 46 | 8 | | Database Block external name |
| 54 | 2 | | Block type |
| 56 | 8 | | External name of the schema |
| 64 | 1 | | Control cards in front of block |
| 65 | 1 | | Control cards in back of block |
| 66 | 4 | | Version number |

Description (Line L2):

'L2' is the line code used to describe a Hierarchical Database Block.

The same line code is used for the Descriptions of SOCRATE/CLIO sub-structures but only the following lines are filled in: the block code, the action code, the line number and, in the column reserved for the Model Relationship code, the code of the structure to which the sub-structure belongs.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L2' | |
| 4 | 6 | | Database Block code |
| 10 | 3 | | Line number |
| 13 | 4 | | Child segment code |
| 17 | 4 | | Parent segment code |
| 21 | 6 | | Model Relation code |
| 27 | 1 | | Identifier in Segment |
| 28 | 5 | | Estimated number: child/parent links |
| | | | Numeric |
| 33 | 36 | | Comment/relation/key length |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 69 | 6 | | Path item (turboimage) |
| 75 | 6 | | Sort path item (turboimage) |

Database Blocks (CodasyI)

Definition (Line L1):

'L1' is the line code used to define a Database Block.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L1' | |
| 4 | 6 | | Database Block code |
| 10 | 36 | | Name of the block |
| 46 | 8 | | Database Block external name |
| 54 | 2 | | Block type |
| 56 | 8 | | External name of the schema |
| 64 | 1 | | Control cards in front of block |
| 65 | 1 | | Control cards in back of block |
| 66 | 4 | | Version number |

Description (Line L3):

'L3' is the line code used to describe CODASYL, DB2, and TANDEM Database Blocks.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L3' | |
| 4 | 6 | | Database Block code |
| 10 | 3 | | Line number |
| 13 | 1 | | TYPE |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | 'S' | Set. |
| | | '*' | Continuation of a set. |
| | | | For a set with multiple members, the first MEMBER Segment is indicated on an 'S'-type line, the others on '*'-type lines. |
| | | 'R' | Record. |
| | | 'A' | Area. |
| 14 | 6 | | Area or set code |
| 20 | 4 | | Parent segment code |
| 24 | 4 | | Child segment code |
| 28 | 6 | | Model Relation code |
| 34 | 5 | | Estimated number: child/parent links |
| | | | Numeric |
| 39 | 36 | | Comment/relation/key length |

Database Blocks (Relational-SQL)

Definition (Line L1):

'L1' is the line code used to define a Database Block.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L1' | |
| 4 | 6 | | Database Block code |
| 10 | 36 | | Name of the block |
| 46 | 8 | | Database Block external name |
| 54 | 2 | | Block type |
| 56 | 8 | | External name of the schema |
| 64 | 1 | | Control cards in front of block |
| 65 | 1 | | Control cards in back of block |
| 66 | 4 | | Version number |

Description (Line L4):

'L4' is the line code used to describe a Relational/SQL Database Block.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L4' | |
| 4 | 6 | | Database Block code |
| 10 | 3 | | Line number |
| 13 | 1 | | Structure code SQL batch transact |
| 14 | 1 | | SQL record type |
| | | 'P' | Tablespace (except for Interrel RDBC, Interrel RFM, Nonstop SQL, Sybase and SQL Server) |
| | | 'T' | Table |
| | | 'V' | View |
| | | 'I' | Index |
| | | 'A' | Alter Table: Column updating |
| | | 'K' | RDMS 1100: Primary Key (Processed with the generation of the table that precedes it.) |
| | | | DB2, Datacom/DB, SQL/DS, Oracle V6 and V7, DB2/2, DB2/6000, Sybase and SQL Server: Primary key (Processed with the generation through an ALTER TABLE command.) |
| | | 'J' | DB2, Datacom/DB, SQL/DS, Oracle V6 and V7, Sybase and SQL Server: Foreign key (Processed with the generation through an ALTER TABLE command.) |
| | | 'C' | Package (Oracle V7 only) |
| | | 'E' | Function (Oracle V7 only) |
| | | 'Q' | Procedure (Oracle V7, Sybase, SQL Server) |
| | | 'R' | Oracle V7, Sybase and SQL Server: Trigger |
| 15 | 18 | | Method external name |
| 33 | 4 | | Segment code |
| 37 | 1 | | Order |
| 41 | 1 | | Key type |
| 43 | 1 | | Type of generated transaction |
| 44 | 6 | | Code of key data element no.1 |
| 50 | 1 | | Sort order 1 |
| 51 | 6 | | Code of key data element no.2 |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 57 | 1 | | Sort order 2 |
| 58 | 6 | | Code of key data element no.3 |
| 64 | 1 | | Sort order 3 |
| 65 | 6 | | Code of key data element no.4 |
| 71 | 1 | | Sort order 4 |
| 72 | 6 | | Code of key data element no.5 |
| 78 | 1 | | Sort order 5 |

Database Blocks (Turboimage)

Definition (Line L1):

'L1' is the line code used to define a Database Block.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L1' | |
| 4 | 6 | | Database Block code |
| 10 | 36 | | Name of the block |
| 46 | 8 | | Database Block external name |
| 54 | 2 | | Block type |
| 56 | 8 | | External name of the schema |
| 64 | 1 | | Control cards in front of block |
| 65 | 1 | | Control cards in back of block |
| 66 | 4 | | Version number |

Description (Line L2):

'L2' is the line code used to describe a Hierarchical Database Block.

The same line code is used for the Descriptions of SOCRATE/CLIO sub-structures but only the following lines are filled in: the block code, the action code, the line number and, in the column reserved for the Model Relationship code, the code of the structure to which the sub-structure belongs.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'L2' | |
| 4 | 6 | | Database Block code |
| 10 | 3 | | Line number |
| 13 | 4 | | Child segment code |
| 17 | 4 | | Parent segment code |
| 21 | 6 | | Model Relation code |
| 27 | 1 | | Identifier in Segment |
| 28 | 5 | | Estimated number: child/parent links |
| | | | Numeric |
| 33 | 36 | | Comment/relation/key length |
| 69 | 6 | | Path item (turboimage) |
| 75 | 6 | | Sort path item (turboimage) |

Texts

Definition (Line S):

'S' is the line code used to define a Text.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'S' | |
| 3 | 6 | | Text code |
| 9 | 36 | | Text name |
| 45 | 2 | | Type of text |
| 47 | 2 | | Paragraph type |

Description (Line T):

'T' is the line code used to describe a text.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction code |
| 2 | 1 | | Line code |
| | | 'T' | |
| 3 | 6 | | Text code |
| 9 | 2 | | Text paragraph |
| 11 | 3 | | Line number |
| 14 | 1 | | Type of Text line |
| | | | Section title: |
| | | | A section must always contain a title. The choice TtttttLT is used to consult the list of titles. |
| | | 'L' | Section title. It will NOT appear in a Volume. |
| | | 'K' | Section title. This line will appear in a Volume. |
| | | '-' | Section title. This line will be underlined with the '-' (dash) character when a Volume is printed. |
| | | '_' | Section title. This line will be underlined with the '_' (underscore) character when a Volume is printed. |
| | | '=' | Section title. This line will be underlined with the '=' character when a Volume is printed. |
| | | '+' | Section title. This line will be underlined with the '+' character when a Volume is printed. |
| | | | Text Description line: |
| | | blank | Description line printed without additional skip (default option). |
| | | ('1' '9') | Number of lines to skip before the line text printing: the value '1' means new line, the value '2' inserts one line before the given line is printed, the value '3' inserts two lines, etc. |
| | | '*' | PAGE skip before the given line is printed. |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | | Text Assignment: Text can be assigned to instances of other entities called in the TEXT DESCRIPTION LINE field. The assignment starts at the beginning of the section which contains the I-type line and terminates at the end of the text or after a J-type line. The assignment for one instances, all instances of a given entity or of all entities can be terminated. The '-AT' choice is used to visualize the texts assigned to the instance of an entity. Texts can be assigned to the following entities: 'B' (Database block), 'D' (Data structure), 'E' (Data element), 'F' (Meta-entity), 'I' (Input Aid), 'M' (Model entity), 'O' (Screen), 'P' (Program), 'Q' (User relation), 'R' (Report), 'S' (Segment), 'T' (Text), 'V' (Volume), '\$' (User entity). |
| | | 'T' | Beginning of assignment. It starts at the beginning of the section which contains this line. |
| | | 'J' | Explicit end of assignment. |
| | | 'B' | Same as type 'T' plus possibility to enter codes of User Entity instances longer than 6 characters. |
| | | 'E' | Same as type 'J' plus possibility to enter codes of User Entity instances longer than 6 characters. |
| | | 'Y' | This code is used to create a link between this section of text and another text or section, i.e. 'refer to..'. The System displays the title of this text or section. |
| | | | For the referenced text: |
| | | | Choice -XT gives the list of texts referring to the whole text, Choice -LT gives the list of sections, each followed by the sections referring to it. |
| | | | Note: The L, I, J, B, E and Y Type lines are not printed in Volumes. |
| 15 | 60 | | Text contents |
| 75 | 6 | | Element code |

Documents

Definition (Line W1):

'W1' is the line code used to define a Document.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 2 | 2 | | Line code |
| | | 'W1' | |
| 4 | 6 | | Volume code |
| 10 | 36 | | Volume name |
| 46 | 1 | | Volume type |
| 47 | 1 | | Title page option |
| 48 | 1 | | Table of contents source |
| 49 | 1 | | Table of contents placement |
| 50 | 6 | | Text code |
| 56 | 3 | | Report Code |
| 59 | 3 | | Report code for font types |
| 62 | 3 | | Report code for specific layout |
| 65 | 1 | | Volume description organization mode |

Description (Line W2):

'W2' is the line code used to describe a Document.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'W2' | |
| 4 | 6 | | Volume code |
| 10 | 2 | | Level 1 code |
| 12 | 2 | | Level 2 code |
| 14 | 3 | | Line number |
| 17 | 1 | | Type of volume description line |
| 18 | 1 | | Section level number |
| 19 | 1 | | Line skip/page break |
| | | | Numeric |
| 20 | 1 | | Character for title underlining |
| 21 | 1 | | Print window |
| 22 | 1 | | Alignment option |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 23 | 50 | | Title, printing opt. or entity sel. |
| 73 | 4 | | Reference cursor |

Parameterized Input Aids

Definition (Line V1):

'V1' is the access line used to define a P.I.A.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'V1' | |
| 4 | 6 | | Input Aid code |
| 10 | 36 | | Parameterized Input Aid name |
| 46 | 1 | | Parameterized Input Aid type |
| | | 'C' | Comment |
| | | 'G' | Generation |
| | | 'O' | Option |

Description (Line V2):

'V2' is the line code used to describe a P.I.A.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'V2' | |
| 4 | 6 | | Input Aid code |
| 10 | 3 | | Line number |
| 13 | 1 | | Line type |
| 14 | 20 | | Label |
| 34 | 29 | | Initial value of P.I.A. line |
| 63 | 3 | | Length |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 66 | 6 | | Reference Name |
| 72 | 1 | | Line Option |

Meta-Entities

Definition (Line Y1):

'Y1' is the line code used to define a Meta-Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'Y1' | |
| 4 | 6 | | Client Meta-Entity code |
| 10 | 36 | | Client Meta-Entity label |
| 46 | 2 | | Meta-Entity calling code |

Detail Line Definition (Line Y6):

'Y6' is the line code used to define the UE detail lines of the Meta-Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| 1 | 1 | | Transaction Code |
| | | SPACE | Creation-Modification |
| | | 'X' | Creation-Modification with data containing an ampersand |
| | | 'C' | Creation |
| | | 'M' | Modification |
| | | 'A' | Deletion |
| | | 'B' | Beginning of multiple deletion |
| | | 'R' | End of multiple deletion |
| | | 'S' | Standard function deletion |
| 2 | 2 | | Line code |
| | | 'Y6' | |
| 4 | 6 | | Client Meta-Entity code |

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 10 | 2 | | Description type |
| 12 | 1 | | Description type |
| 13 | 30 | | Me. description label |
| 43 | 8 | | Subprogram code |
| 51 | 1 | | Data storage mode |
| 54 | 2 | | Parent description type |

Description (Line Y2):

'Y2' is the line code used to describe a Meta-Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'Y2' | |
| 4 | 6 | | Client Meta-Entity code |
| 10 | 2 | | Description type |
| 12 | 3 | | Line number |
| 15 | 6 | | Element code |
| 21 | 2 | | Range |
| 23 | 1 | | Element top nature |
| 24 | 1 | | Uppercase top change |
| 25 | 1 | | Element format top control |
| 26 | 1 | | Presence top control |
| 27 | 1 | | Value top control |
| 28 | 6 | | User Relation Code |
| 73 | 1 | | Parent identifier code |
| 74 | 1 | | Identifier code called by Relation |

User-Defined Relations

Definition (Line Y5):

'Y5' is the line code used to define a User-Defined Relation.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'Y5' | |
| 4 | 6 | | Client User Relation |
| 10 | 36 | | Client User Relation label |
| 46 | 14 | | Client User Relation short label |
| 60 | 1 | | Client User Relation type |
| 61 | 3 | | Entity Type (3 characters) |
| | | | The authorized values are the Entity type values given in chapter "DAF Entities: Coding rules", subchapter "Tables" of the "DSMS Access Facility Tables" manual. |
| 64 | 1 | | Deletion flag |

Client User Entities

Definition (Line Y3):

'Y3' is the line code used to define a Client User Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'Y3' | |
| 4 | 2 | | Meta-Entity calling code |
| 6 | 6 | | User Entity short code |
| 12 | 2 | | Range |
| 14 | 1 | | Transaction number for User Entity |
| | | blank | First line |
| | | '*' | Continuation line |
| 15 | 66 | | User Entity Definition Transaction |

Description (Line Y4):

'Y4' is the line code used to describe the detail lines of a Client User Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'Y4' | |
| 4 | 2 | | Description code |
| 6 | 6 | | User Entity descr. short identifier |
| 12 | 2 | | Range |
| 14 | 1 | | Transaction number for User Entity |
| | | blank | First line |
| | | '*' | Continuation line |
| 15 | 66 | | User Entity Definition Transaction |

Extension User Entities

Definition (Line YC):

'YC' is the line code used to define an Extension User Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'YC' | |
| 4 | 2 | | Meta-Entity calling code |
| 6 | 6 | | User Entity short code |
| 12 | 2 | | Range |
| 14 | 1 | | Transaction number for User Entity |
| | | blank | First line |
| | | '*' | Continuation line |
| 15 | 66 | | User Entity Definition Transaction |

Description (Line YD):

'YD' is the line code used to describe the detail lines of an Extension User Entity.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'YD' | |
| 4 | 2 | | Description code |
| 6 | 6 | | User Entity descr. short identifier |
| 12 | 2 | | Range |
| 14 | 1 | | Transaction number for User Entity |
| | | blank | First line |
| | | '*' | Continuation line |
| 15 | 66 | | User Entity Definition Transaction |

Thesaurus

Enrichment of the Thesaurus (Line G1):

'G1' is the access line used to document keywords (enrichment of the Thesaurus).

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 1 | | Transaction code |
| 2 | 2 | | Line code |
| | | 'G1' | |
| 4 | 13 | | KEYWORD |
| 17 | 1 | | Continuation line |
| 18 | 1 | | Keyword description type |
| | | 'D' | Comments |
| | | 'S' | Synonym(s) |
| 19 | 55 | | Keyword description |

Library

Definition (Line X):

'X' is the line code used to define a Library.

| POS | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|--|
| 1 | 36 | | Library name |
| 37 | 1 | | Date format indicator |
| 38 | 1 | | Vertical character of frame |
| 39 | 1 | | Stream OCLS/BEFORE |
| 40 | 1 | | Stream OCLS/AFTER |
| 41 | 2 | NUMER. | Lines per page in documentation |
| | | | Numeric |
| 43 | 1 | | Page skip |
| 44 | 1 | | Comments Insertion Option |
| 45 | 1 | | Modification of extracted lines |
| 46 | 1 | | Optional Command Lines Set/BEFORE |
| 47 | 1 | | Optional Command Lines Set/AFTER |
| 48 | 1 | | Generation Language |
| 49 | 1 | | Type of COBOL |
| 50 | 1 | | Programming mode |
| 51 | 1 | | Protection of extracted entities |
| 57 | 1 | | Date Format in Generated programs |
| 58 | 1 | | Decimal Point Presentation character |
| 59 | 1 | | TP Monitor and Map Type |
| 60 | 1 | | Generated COBOL formatting |
| 61 | 1 | | Alphanumeric Delimiter |
| 62 | 1 | | Horizontal character of frame |
| 63 | 1 | | Century System Date |
| 64 | 2 | NUMER. | Reference Year for Century |

UPDT - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Database consistency check: PTUBAS

| Code | Physical name | Type | Label |
|--------|-----------------|-------|---------------------------|
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database data |

| Code | Physical name | Type | Label |
|--------|---------------|--------|-----------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAC7DS | | Report | Validity report |

Return code:

This utility sends a return code 12 and the procedure stops running if the database is invalid.

Transactions formatting: PACA05

| Code | Physical name | Type | Label |
|--------|------------------|--------|---|
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7AY | &INDUV..&BASE.AY | Input | Development Database extension data |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7MB | &&UPDTMB | Input | Update transactions |
| PAC7ME | &&PAC7ME | Output | Work file (length=372) |
| PAC7MV | &&PAC7MV | Output | Formatted transactions (length=170, must be able to contain all input transactions plus the elementary delete transactions generated by the multiple delete transactions) |
| PAC7MW | | Output | Work file |

Update of the Development Database: PACA15

| Code | Physical name | Type | Label |
|--------|------------------|--------|--------------------------------|
| PAC7AR | &INDUV..&BASE.AR | Output | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Output | Development Database index |
| PAC7AY | &INDUV..&BASE.AY | Output | Development Database extension |
| PAC7AJ | &INDUV..&BASE.AJ | Output | Development Database journal |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |

| Code | Physical name | Type | Label |
|--------|---------------|--------|---|
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGY | &INDSV..BVPGY | Input | Administration Database Extension |
| PACGGU | &INDSV..BVPGU | Input | Administration Database users |
| PAC7DC | &DSMS | Input | DSMS file of Development Database Elements |
| PAC7ME | &&PAC7ME | Input | Work file |
| PAC7MV | &&PAC7MV | Input | Update transactions |
| PAC7RB | &&PAC7RB | Output | UPDT erroneous transactions (length=80) |
| PAC7RY | &&PAC7RY | Output | UPDP erroneous transactions (length=310) |
| PAC7IE | | Report | Update report (length=132) |
| PAC7IF | | Report | List of erroneous transactions (length=132) |

The list of user transactions is preceded by a banner with the user code.

Return codes:

- 0: OK, no error
- 2: Warning
- 4: Critical error

UPDT - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**              - BATCH UPDATE -
/**
/** -----
/**
/** REFER TO THE BATCH FORMS AND TO THE DESCRIPTION OF THE
/** INPUT CORRESPONDING TO EACH ENTITY.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/**   COL 2 : "*"
/**   COL 3 : USERIDXX
/**   COL 11 : PASSWORD
/**   COL 28 : LANGUAGE CODE, USEFUL WHEN TRANSACTION ARE

```

```

/**          NOT IN THE SAME LANGUAGE AS THE DATABASE.
/** COL 67 : "N"  DEFAULT VALUE WITH EXTRACTORS
/** - COMMAND LINE
/** THE LIST OF ALL AVAILABLE VALUES FOR THE ENTITY
/** TO BE UPDATED IS FOUND IN REFERENCE MANUAL.
/**
/** -----
/**
//BVPUPDT  PROC BASE=$BASE,          CODE OF DEVPT DATABASE
//          INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//          INDUV='$INDUV',          INDEX OF USER VSAM FILES
//          DSMS='$DSMS',           DSNAME OF PRODUCT ELEMENTS DSMS FILE
/**:      VSAMCAT='$VCAT',           USER VSAM CATALOG
/**:      SYSTCAT='$SCAT',           SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          LSR='BLSR',              LSR BATCH SYSTEM NAME
//          OUT=$OUT,                EXECUTION ERRORS OUTPUT CLASS
//          OUTL=$OUT,               OUTPUT CLASS OF REPORTS
//          UWK=$UWK,                WORK UNIT
//          SPAMB='(TRK,(100,10),RLSE) ' TRANSACTION SPACE
/**-----
//INPUT EXEC PGM=BVPTU001
/**-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&UPDTMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
/**-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**:      DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AJ  DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AY  DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGY  DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTUBAS EXEC PGM=BVPTUBAS
/**-----
/**:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
/**:      DD DSN=&VSAMCAT,DISP=SHR

```



```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7DS DD SYSOUT=&OUT
//PACA05 EXEC PGM=BVPACA05,COND=(0,NE,PTUBAS)
//*-----
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7MB DD DSN=&&UPDTMB,DISP=(OLD,DELETE)
//PAC7ME DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=372,BLKSIZE=5952)
//PAC7MV DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW DD DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PACA15 EXEC PGM=BVPACA15,COND=((0,NE,PACA05),(0,NE,PTUBAS))
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')

```

```

//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC DD DSN=&DSMS,DISP=SHR
//PAC7IE DD SYSOUT=&OUTL
//PAC7IF DD SYSOUT=&OUTL
//PAC7ME DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB DD DSN=&&PAC7RB,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PAC7RY DD DSN=&&PAC7RY,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)
//SYSUDUMP DD SYSOUT=&OUT

```

Chapter 6. Pactables

GETD-GETA - Description Generators

GETD-GETA - Introduction

The Table Description Generator is the interface between the Specifications Dictionary and Pactables. For further information, refer to chapter 'General Introduction', subchapter "Introduction to the Pactables Facility" in the "Pactables" manual.

Its use is subject to a purchase agreement.

This interface extracts, from the VisualAge Pacbase Database, the descriptions of the Tables necessary for the operation of the Pactables Facility.

This extraction is executed via either the GETA or GETD procedure according to the installation environment of the Pactables Facility:

- GETA if the Dictionary and Pactables are running under the same environment.
- GETD if the Dictionary and Pactables are running under different environments. In this case, GETD processes a table description file which is the image of the file containing the table descriptions used by the Pactables Facility. As a result, this file must be initialized before the first GETD run, by:
 - either duplicating the description file of the Pactables Facility, if it exists,
 - or executing the initialization procedure (GETI) described in this chapter.

GETA or GETD provides an interface file which is used as input to the GETT procedure of the Pactables Facility. For further details, refer to the 'Pactables' manual.

Execution conditions

None with regard to the Specifications Database, which is only read by this procedure.

Abnormal execution

If the generation abends before the update of the table description file, the procedure can be restarted as it is once the error has been corrected.

If the generation abends during the update of the table description file, this file must be restored before the procedure is restarted.

GETD-GETA - User Input / Result

User input

A '*'-type line indicating the Library which contains the table descriptions.

| Position | Length | Value | Meaning |
|----------|--------|----------|----------------|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |
| 19 | 3 | bbb | Library code |
| 22 | 4 | nnnn | Session number |
| 26 | 1 | t | Session status |

One 'Z' line per generation or print request.

| Position | Length | Value | Meaning |
|----------|--------|----------|--|
| 2 | 1 | 'Z' | Line code |
| 5 | 4 | | Request code: |
| | | 'TGS ' | Request for table descrip. generation |
| | | 'TDS ' | Request for printing of table descr. |
| | | 'TLS ' | Request for list of table descriptions |
| | | 'TAS ' | Request for table deletion |
| | | 'TMS ' | Request for modification of frozen table characteristics |
| | | 'TGC ' | Request for comments generation |
| 9 | 6 | ssss | Segment code of table description to be extracted ('TGS ','TGC ') |
| | | ttttt | Table code (other requests) |
| 15 | 2 | ' ' | Not significant |
| 17 | 8 | MMDDCCYY | Date from which the table description can be modified. (Optional) |
| 25 | 8 | MMDDCCYY | Date of description historical account for a G-type table. Default: last historical account. |
| | | ***** | Table generation without hist. account |
| 33 | 1 | | Data Element format type: |

| Position | Length | Value | Meaning |
|----------|--------|--------|---|
| | | ' ' | Internal format |
| | | 'E' | Input format |
| 75 | 6 | tttttt | Table number (if generating for a table other than that of the Segment's Definition in the Database). |

For further information on this user input, please refer to the Pactables Reference Manual.

Note: Table keys cannot be modified. The generation requests for tables already defined and which involve such modifications are rejected.

Result

The output of the GETA procedure is a sequential file containing table descriptions, which will be used as input to the GETT procedure of the Pactables Facility.

GETD-GETA - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Extraction & update preparation: PACT40

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--|
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV.&BASE.AN | Input | Development Database Index file |
| PAC7AY | &INDUV.&BASE.AY | Input | Development Database extension data |
| PACGGN | &INDSV..BVPGN | Input | Administration Database index |
| PACGGR | &INDSV..BVPGR | Input | Administration Database data |
| PACGGU | &INDSV..BVPGU | Input | Administration Database users |
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAC7TD | &TABTDF | Input | Tables descriptions |
| PAC7MB | &&GETDMB or &&GETAMB | Input | Requests for descriptions |
| PAC7MD | &MD | Output | Update transactions for the Descriptions whose version is equal to or greater than 2.0 |
| PAC7ET | | Report | Output report |

| Code | Physical name | Type | Label |
|--------|---------------|--------|--------------------------------------|
| PAC7DD | | Report | Batch procedure authorization option |

Return Codes :

- 8 : no batch procedure authorization.

Formatting of descriptions < V 2.0: PACT45

| Code | Physical name | Type | Label |
|--------|---------------|--------|--|
| PAC7MD | &MD | Input | Update transactions for the Descriptions whose version is equal to or greater than 2.0 |
| PAC7ND | &ND | Output | Update transactions for the Descriptions whose version is lower than 2.0 |

Update of table descriptions file: PACT50

(GETD procedure only)

| Code | Physical name | Type | Label |
|----------|-----------------|--------|--------------------------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database Data file |
| PAC7TD | &TABTDF | Input | Tables descriptions |
| PAC7MD | &MD | Input | Update transactions |
| PAC7ET | | Report | Update report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

GETD - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**          - TABLE DESCRIPTIONS GENERATION -
/**
/** -----
/**
/** TABLE DESCRIPTION GENERATOR IS THE INTERFACE BETWEEN
/** THE SPECIFICATIONS DICTIONARY AND VA PACTABLES.
/** FOR FURTHER INFORMATION, REFER TO CHAPTER GENERAL
/** INTRODUCTION SUBCHAPTER INTRODUCTION TO THE

```

```

/** VA PACTABLES FACILITY IN THE VA PACTABLES MANUAL.
/**
/** GETD IF THE DICTIONARY AND VA PACTABLES ARE RUNNING
/** UNDER DIFFERENT ENVIRONMENTS.
/**
/** -----
/**
/**BVPGETD  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**          TABTDF='$TABTDF',              DSN OF TABLES DESCRIPTION FILE
/**          INDUV='$INDUV',                INDEX OF USER VSAM FILES
/**          INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES
/**          INDSN='$INDSN',                INDEX OF NON-VSAM SYSTEM FILES
/***:       VSAMCAT='$VCAT',                USER VSAM CATALOG
/***:       SYSCAT='$SCAT',                SYSTEM VSAM CATALOG
/**          STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
/**          OUT=$OUT,                      OUTPUT CLASS OF REPORTS
/**          UWK=$UWK,                      WORK UNIT
/**          MD=&MD,                        GENERATED DESCRIPTION FOR PACTABLES >=2.0
/**          ND=&ND,                        GENERATED DESCRIPTION FOR PACTABLE <2.0
/**          UNITS=$UNITUN,                 GENERATED DESCRIPTION UNIT
/**          VOLS='$SER=$VOLUN',            GENERATED DESCRIPTION VOLUME
/**          SORTLIB='$BIBT',              SORT LIBRARY
/**          CYL=2,                        SIZE OF WORK OF SORT
/**          SPAMD='(TRK,(10,2),RLSE)',    SPACE GENERATED DESCRIPTION
/**          SPAND='(TRK,(10,2),RLSE)'     SPACE GENERATED DESCR < 2.0
/*******
/**INPUT  EXEC  PGM=BVPTU001
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**          DD DSN=$BCOB,DISP=SHR
/**CARTE   DD DDNAME=SYSIN
/**PAC7MB  DD DSN=&&GETDMB,DISP=(,PASS),
/**          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
/**          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**VERIFY EXEC PGM=IDCAMS
/**-----
/***:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/***:       DD DSN=&SYSCAT,DISP=SHR
/**SYSPRINT DD SYSOUT=&OUT
/**PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
/**PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
/**PAC7AY  DD DSN=&INDUV..&BASE.AY,DISP=SHR
/**PAC7TD  DD DSN=&TABTDF,DISP=SHR
/**PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
/**PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
/**PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
/**SYSIN   DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
/**          DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
/**          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
/**          DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
/**          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
/**          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
/**          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
/**PACT40  EXEC  PGM=BVPACT40
/**-----

```

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7ET DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&GETDMB,DISP=(OLD,PASS)
//PAC7MD DD DSN=&MD,DISP=(,PASS),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAMD,
// DCB=(RECFM=FB,LRECL=244,BLKSIZE=6100)
//PAC7TD DD DSN=&TABTDF,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT
//PACT45 EXEC PGM=BVPACT45
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7MD DD DSN=&MD,DISP=(OLD,PASS)
//PAC7ND DD DSN=&ND,DISP=(,PASS),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAND,
// DCB=(RECFM=FB,LRECL=244,BLKSIZE=6100)
//SYSUDUMP DD SYSOUT=&OUT
//PACT50 EXEC PGM=BVPACT50,COND=(0,NE,PACT40)
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7ET DD SYSOUT=&OUT
//PAC7MD DD DSN=&MD,DISP=(OLD,CATLG)
//PAC7TD DD DSN=&TABTDF,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```


GETA - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - TABLES DESCRIPTION GENERATION -
/**
/** -----
/**
/** TABLE DESCRIPTION GENERATOR IS THE INTERFACE BETWEEN
/** THE SPECIFICATIONS DICTIONARY AND VA PACTABLES.
/** FOR FURTHER INFORMATION, REFER TO CHAPTER GENERAL
/** INTRODUCTION SUBCHAPTER INTRODUCTION TO THE
/** VA PACTABLES FACILITY IN THE VA PACTABLES MANUAL.
/**
/** GETA IF THE DICTIONARY AND VA PACTABLES ARE RUNNING
/** UNDER THE SAME ENVIRONMENTS.
/**
/** -----
/**
/**BVPGETA PROC BASE=$BASE,          CODE OF DEVPT DATABASE
/**      TABTDF='$TABTDF',          DSN OF TABLES DESCRIPTION FILE
/**      INDUV='$INDUV',            INDEX OF USER VSAM FILE
/**      INDSV='$INDSV',            INDEX OF VSAM SYSTEM FILES
/**      INDSN='$INDSN',            INDEX OF NON VSAM SYSTEM FILES
/***:    VSAMCAT='$VCAT',            USER VSAM CATALOG
/***:    SYSTCAT='$SCAT',            SYSTEM VSAM CATALOG
/**      STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
/**      OUT=$OUT,                  OUTPUT CLASS OF REPORTS
/**      UWK=$UWK,                  WORK UNIT
/**      MD='&&MD',                  GENERATED DESCRIPTION FOR PACTABLES >=2.0
/**      ND='&&ND',                  GENERATED DESCRIPTION FOR PACTABLE <2.0
/**      UNITS=$UWK,                GENERATED DESCRIPTION UNIT
/**      VOLS=,                     GENERATED DESCRIPTION VOLUME
/**      SPAMD='(TRK,(10,2),RLSE)', SPACE GENERATED DESCRIPTION
/**      SPAND='(TRK,(10,2),RLSE)'  SPACE < 2.0 DESCRIPTION
/*******
/**INPUT EXEC PGM=BVPTU001
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**      DD DSN=$BCOB,DISP=SHR
/**CARTE DD DDNAME=SYSIN
/**PAC7MB DD DSN=&&GETAMB,DISP=(,PASS),
/**      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
/**      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**VERIFY EXEC PGM=IDCAMS
/**-----
/***:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/***:      DD DSN=&SYSTCAT,DISP=SHR
/**SYSPRINT DD SYSOUT=&OUT
/**PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
/**PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
/**PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
/**PAC7TD DD DSN=&TABTDF,DISP=SHR

```

```

//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFTD),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PACT40 EXEC PGM=BVPACT40
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCL DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7ET DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&GETAMB,DISP=(OLD,DELETE)
//PAC7MD DD DSN=&MD,DISP=(,PASS),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAMD,
// DCB=(RECFM=FB,LRECL=244,BLKSIZE=6100)
//PAC7TD DD DSN=&TABTDF,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT
//PACT45 EXEC PGM=BVPACT45
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCL DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7MD DD DSN=&MD,DISP=(OLD,PASS)
//PAC7ND DD DSN=&ND,DISP=(,PASS),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAND,
// DCB=(RECFM=FB,LRECL=244,BLKSIZE=6100)
//SYSUDUMP DD SYSOUT=&OUT

```

GETI - Initialization of Description Line

GETI - Introduction

The GETI procedure must be executed when first using Pactables files that are stored in an environment other than the VisualAge Pacbase environment. It initializes the description file in a way similar to the Pactables INTA procedure does.

GETI - User Input

An '*' line with a user code and password.

An 'I' line with initialization parameters.

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 2 | 1 | 'I' | Line code |
| 3 | 36 | | Installation name |
| 39 | 1 | | Language code |
| | | 'F' | French (default option) |
| | | 'E' | English |
| 53 | 4 | cccc | Class for Security System |
| 57 | 1 | | Type of Security System |
| | | 'R' | RACF |
| | | 'S' | Top secret |
| 58 | 2 | nn | Number of lines per printed page |
| 60 | 1 | | Type of resource controls |
| | | ' ' | Definition of Security system tables resources |
| | | 'P' | Definition of resources in the Development Database |
| 61 | 1 | | User code lock |
| | | ' ' | other user code authorized |
| | | 'N' | other user code not authorized |

GETI - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Initialization of descriptions file: PACTIN

| Code | Physical name | Type | Label |
|--------|-----------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV.&BASE.AR | Input | Development Database Data file |
| PAC7MB | &&GETIMB | Input | Parameter line |
| PAC7TD | &TABTDF | Output | Table description file |
| PAC7ED | | Report | Initialization report |
| PAC7DD | | Report | Authorization control |

GETI - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - INITIALIZATION OF TABLES MANAGEMENT FILE -
/**
/** -----
/**
/** THE GETI PROCEDURE MUST BE EXECUTED WHEN FIRST USING
/** VA PACTABLES FILES THAT ARE STORED IN ANOTHER
/** ENVIRONMENT FROM THE PRODUCT ENVIRONMENT.
/** IT INITIALIZES THE DESCRIPTION FILE IN A SIMILAR WAY
/** AS THE VA PACTABLES INTA PROCEDURE DOES.
/**
/** -----
/**
//BVPGETI  PROC BASE=$BASE,           CODE OF DEVPT DATABASE
//          TABTDF='$TABTDF',         DSN OF TABLES DESCRIPTION FILE
//          INDSN='$INDSN',           INDEX OF NON-VSAM FILES
//          INDSV='$INDSV',           INDEX OF SYSTEM VSAM FILES
//          INDUV='$INDUV',           INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',           USER VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          LSR='BLSR',               LSR BATCH SYSTEM NAME
//          UWK=$UWK,                 WORK UNIT
//          OUT=$OUT                   OUTPUT CLASS
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&GETIMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS

```

```

//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//        DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//        DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(DFTABTDF),DISP=SHR
//PACTIN EXEC PGM=BVPACTIN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//GNLSR   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR   DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//ARLSR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7MB   DD DSN=&&GETIMB,DISP=SHR
//PAC7TD   DD DSN=&TABTDF,DISP=SHR
//PAC7ED   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

SMTD-RMTD - Migration of Tables Descriptions

SMTD - Introduction

The SMTD procedure backs up the TD table-description file by transforming binary characters into their display format.

The aim of the procedure is to transfer the TD file onto different platforms while avoiding problems caused by the presence of these characters at the time of transfers.

Execution condition

None.

SMTD - Description of Steps

Input recognition: PTU001

TD Backup (Tables description file): PTASVD

| Code | Physical name | Type | Label |
|--------|---------------|--------|--|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7MB | &&SMTDMB | Input | User transactions |
| PAC7TC | &TDSAV | Output | Table description backup for the migration |
| PAC7TD | &TABTDF | Input | Tables description file |
| PAC7DD | | Report | Authorization control |

SMTD - Execution JCL

```
/* -----  
/*      VISUALAGE PACBASE  
/*  
/* -----  
/*      - BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION -  
/*  
/* -----  
/*  
/* THE SMTD PROCEDURE BACKS UP THE TD TABLE-DESCRIPTION  
/* FILE BY TRANSFORMING BINARY CHARACTERS INTO THEIR  
/* DISPLAY FORMAT.  
/*  
/* -----  
/*  
//BVPSMTD PROC TABTDF='$TABTDF', DSN OF TABLES DESCRIPTION FILE  
//      TDSAV=,          DSN OF TABLES DESCRIPTION BACKUP FILE  
//      INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES  
//*:      VSAMCAT='$VCAT',          VSAM USER CATALOG  
//      STEPLIB='$HLQ..SBVPMBR8',          LIBRARY OF LOAD-MODULES  
//      UWK=$UWK,          WORK UNIT  
//      OUT=$OUT,          OUTPUT CLASS  
//      VOL=$SER=$VOLUN',          BACKUP VOLUME  
//      UNITS=$UNITUN,          BACKUP UNIT (DISK OR CARTRIDGE)  
//      SPATD='(TRK,(150,10))'          BACKUP SPACE  
//*****  
//INPUT EXEC PGM=BVPTU001  
//*-----  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//      DD DSN=$BCOB,DISP=SHR  
//CARTE DD DDNAME=SYSIN
```

```

//PAC7MB DD DSN=&&SMTDMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PTASVD EXEC PGM=BVPTASVD
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//PACGGN DD DSN=&INDSV..BVP GN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVP GR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7TD DD DSN=&TABTDF,
//          DISP=SHR
//PAC7TC DD DSN=&TDSAV,
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          DISP=(,CATLG,DELETE),SPACE=&SPATD,
//          DCB=(RECFM=VB,LRECL=1067,BLKSIZE=10674)
//PAC7MB DD DSN=&&SMTDMB,DISP=(OLD,DELETE)
//PAC7DD DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

RMTD - Introduction

The Restoration of Table Descriptions procedure (RMTD) restores the TD file of Table Descriptions from its TC sequential backup produced by the SMTD procedure.

This procedure does not require any specific execution condition.

RMTD - Description of Steps

Input recognition: PTU001

TD File Restoration: PTARSD

| Code | Physical name | Type | Label |
|--------|----------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7MB | &&RMTDMB | Input | User transactions |
| PAC7TD | &TDSAV | Output | Tables Description file |
| PAC7TC | &TABTDF | Input | Tables Description sequential file |
| PAC7DD | | Report | Authorization control |

RMTD - Execution JCL

```
//* -----
//*      VISUALAGE PACBASE
//*
//* -----
//*      - RESTORATION OF TABLE DESCRIPTIONS -
//*
//* -----
//*
//* THE RESTORATION OF TABLE DESCRIPTIONS PROCEDURE
//* (RMTD)   RESTORES THE TD FILE OF
//* TABLE DESCRIPTIONS FROM ITS TC SEQUENTIAL BACKUP
//* PRODUCED BY THE SMTD PROCEDURE.
//*
//* -----
//*
//BVPRTD  PROC TABTDF='$TABTDF', DSN OF TABLES DESCRIPTION FILE
//        TDSAV=,           DSN OF TABLES DESCRIPTION BACKUP FILE
//        INDSV='$INDSV',   INDEX OF SYSTEM VSAM FILES
//        INDSN='$INDSN',   NON VSAM SYSTEM FILES INDEX
//*:      VSAMCAT='$VCAT',   VSAM USER CATALOG
//        STEPLIB='$HLQ..SBVPMBR8',   LOAD MODULES LIBRARY
//        UWK=$UWK,         WORK UNIT
//        OUT=$OUT         OUTPUT CLASS
//*****
//INPUT  EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//CARTE  DD DDNAME=SYSIN
//PAC7MB DD DSN=&&RMTDMB,DISP=(,PASS),
//        UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//DELDEF EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN   DD DSN=&INDSN..BVPSY(DFTABTDF),DISP=SHR
//PTARSD  EXEC PGM=BVPTARSD
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7AE  DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7TC  DD DSN=&TDSAV,DISP=OLD
//PAC7TD  DD DSN=&TABTDF,DISP=SHR
//PAC7MB  DD DSN=&&RMTDMB,DISP=(OLD,DELETE)
//PAC7DD  DD SYSOUT=&OUT
//SYSOUT  DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Chapter 7. Pac/Impact

Foreword

Note: Pac/Impact users may also refer to the 'Pac/Impact for VisualAge Pacbase' manual.

Warning

An Impact analysis requires a very large amount of machine-time. It is therefore recommended to limit the scope of the analysis.

You can limit your analysis to two distinct levels. You can also combine two levels, to define a more precise analysis domain.

- The UXSR procedure, documented in 'The Administrator's Procedures' manual, allows you to create a new image of the Development Database, by zooming on a given sub-network (the session extraction is also available). This creates a new Database which is a subset (restructured and/or renamed) of the initial Database. The analysis is then performed on this subset.

Furthermore, the REOR procedure (which must always be run after a UXSR) allows you to cancel instances which are not relevant to the analysis.

- You may also choose to limit your analysis to some instances of the Program, Screen or Database Block entities. Additional selection options are available to this effect.

This analysis limitation is performed by the INFP utility, documented in this manual.

- The procedures in this Function do not impact the Database files. However, it is recommended to close the on-line files for better performance.

INFP - FP File Initialization (Impact Analysis)

INFP - Introduction

This procedure allows to specify the entities which are to be analyzed and thus to narrow the scope of the impact analysis.

For the FP file to be updated by INFP, you must re-enter, in the procedure's input, all the lines already entered. You always start with an empty file, i.e. a file which contains no particular selection.

Result

The procedure outputs a file which contains the entities selected for the analysis (FP).

INFP - User Input

A '*' line with the user code and password.

Other input is optional, knowing that if no input is provided, all the entities of all entity types will be searched for in the impact analysis.

If you request all the existing entities of a given entity type (code = *****), you cannot indicate any specific entities for this type.

If you specify a type in an input line (whether or not you specify an entity for this type), you must also specify, on additional input lines, all the other types to be analyzed by the procedure.

| Position | Length | Value | Meaning |
|----------|--------|--------|--|
| 1 | 3 | | Entity type Possible values are: |
| | | 'B ' | Database Blocks |
| | | 'F ' | Meta-Entities |
| | | 'O ' | Screens |
| | | 'P ' | Programs |
| | | 'T ' | Texts |
| | | 'V ' | Documents |
| | | \$nn | User Entities of 'nn' type code. |
| | | '\$**' | All UEs |
| 4 | 6 | | Entity code (generic selection through code *****) (This code may not exist in the Database) |

INFP - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Preparation of FP KSDS file's DELETE/DEFINE: PRMSYS

This program parameterizes the FP KSDS file name. The file's DELETE/DEFINE is found in the DF&BASE.FP member of the SY-parameters PDS.

The program is followed by an IDCAMS, which performs the DELETE/DEFINE of the FP KSDS file.

| Code | Physical name | Type | Label |
|--------|------------------------------|--------|--|
| PACRIN | &INDUN..&BASE.SY(DF&BASE.FP) | Input | DELETE/DEFINE skeleton of the file |
| PACROU | &&DFFP | Output | DELETE/DEFINE of the file (SYSIN for IDCAMS) |

Definition of the KSDS work file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DFFP | Input | DEFINE file |

Check on transactions and FP update: PAN205

| Code | Physical name | Type | Label |
|----------|-------------------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7MB | &&INFPMB | Input | User input |
| PAC7FP | &INDUV..&USER..&BASE.FP | Output | Entities in production |
| PAC7IP | | Report | Validation report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK.
- 12 : System error.

INFP - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - IMPACT ANALYSIS:  INITIALIZATION OF "FP" FILE -
/**

```

```

/** -----
/**
/** THE INFP PROCEDURE INITIALIZES THE FP FILE. IT ALLOWS
/** TO SPECIFY THE ENTITIES WHICH ARE TO BE ANALYZED AND
/** THUS TO NARROW THE SCOPE OF THE IMPACT ANALYSIS TO SOME
/** (OR ALL) OCCURRENCES OF THE ENTITIES.
/**
/** -----
/**
//BVPINFP  PROC BASE=$BASE,           CODE OF DEVPT DATABASE
//          USER=,                   PACKAGE CODE FOR IMPACT ANALYSIS
//          INDSV='$INDSV',           INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',           INDEX OF SYSTEM NON-VSAM FILES
//          INDUV='$INDUV',           INDEX OF USER VSAM FILES
//          INDUN='$INDUN',           INDEX OF USER NON-VSAM FILES
/**:       VSAMCAT='$VCAT',           USER VSAM CATALOG
/**:       SYSTCAT='$SCAT',           SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',           SORT LIBRARY
//          LSR='BLSR',                LSR BATCH SYSTEM NAME
//          UWK=$UWK,                  WORK UNIT
//          CYL=1,                     SIZE OF SORT FILE
//          OUT=$OUT                   OUTPUT CLASS
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&INFPMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDUN..&BASE.SY(DF&BASE.FP),DISP=SHR
//PACROU DD DSN=&&DFFP,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
/**:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFFP,DISP=(OLD,DELETE)
/**
//VERIFY EXEC PGM=IDCAMS
//*-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR

```

```

//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN205 EXEC PGM=BVPAN205
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FP DD DSN=&INDUV..&USER..&BASE.FP,DISP=SHR
//PAC7IP DD SYSOUT=&OUT
//PAC7MB DD DSN=&&INFPMB,DISP=(OLD,DELETE)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

ISEP - Selection of Entry Points

ISEP - Introduction

The ISEP procedure is designed to select the entry points -- Elements and/or character strings -- which will be used as criteria by the impact analysis (IANA procedure).

The identification line of the selection context (* line) is required. It allows you to specify the session and the sub-network (view Z1) from which the selection will be made.

Elements and character strings are considered as entry points when they meet the selection criteria entered in ISEP user input lines.

Three types of criteria may be used (see below) and at least one selection criterion is required, knowing that no particular criterion type is required.

A selection may combine several types of criteria, and several command lines for each type.

- The E-type line allows you to extract Elements by selecting a code (generic code authorized) and/or one or several format(s).
- The S-type line allows you to extract character strings by selecting a code (generic code authorized) and/or one or several format(s).
- The W-type line allows you to select Elements via a keyword. You may also indicate the keyword type, Element formats and code.

Execution conditions

None.

Abnormal execution

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

Result

Output of the ISEP procedure is two files which are to be used in the IANA procedure:

- 'FH' file which contains the selected entry points,
- 'FR' file which contains the entry points to be purged.

ISEP - User Input

Only one '*' line (required, located at the beginning of the stream):

| Position | Length | Value | Meaning |
|----------|--------|----------|--|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | Password |
| 19 | 3 | bbb | Code of the highest library in the sub-network |
| 22 | 4 | ssss | Session number (blank if current session) |
| 26 | 1 | | Session status (' ' or 'T') |
| 69 | 3 | iii | Code of the lowest Library in the sub-network (optional) |

One E-type line: Selection of Elements (optional):

| Position | Length | Value | Meaning |
|----------|--------|-------|-----------|
| 2 | 1 | 'E' | Line code |

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 3 | 6 | | Element code (generic code possible with the '*' character, at beginning or end of code: ***XXX or XXX**, or with the '?' character followed by the string to be included in the code (?XXX)) |
| 9 | 10 | | Element input format |
| 19 | 10 | | Element internal format |
| 29 | 1 | | Internal usage (default: D) |
| 30 | 27 | | Element output format |
| 57 | 1 | 'N' | Child Elements not impacted |
| | | ' ' | Child Elements impacted |

One S-type line: Selection of character strings (optional)

| Position | Length | Value | Meaning |
|----------|--------|-------|--|
| 2 | 1 | 'S' | Line code |
| 3 | 30 | | String code (generic code possible with the '*' character anywhere in the code), or ?xx where xx is a string located anywhere in the sequence of char. |
| 33 | 10 | | Internal format of the string |
| 43 | 1 | | Internal usage (Default: D) |

One W-type line: Selection on keyword (optional)

| Position | Length | Value | Meaning |
|----------|--------|-------|--|
| 2 | 1 | 'W' | Line code |
| 3 | 1 | | Keyword type (implicit 'L', explicit 'M', or both ' ') |
| 4 | 13 | | Keyword code (no generic code) |
| 17 | 10 | | Element input format |
| 27 | 10 | | Element internal format |
| 37 | 1 | | Internal usage (Default: D) |
| 38 | 27 | | Element output format |
| 65 | 6 | | Element code (generic code possible with the '*' character anywhere in the code) |

| Position | Length | Value | Meaning |
|----------|--------|-------|-----------------------------|
| 71 | 1 | 'N' | Child Elements not impacted |
| | | ' ' | Child Elements impacted |

ISEP - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Selection of entry points: PAN210

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7FP | &INDUV..&USER..&BASE.FP | Input | File of entities to be analyzed |
| PAC7MB | &&ISEPMB | Input | User input |
| PAC7FH | &&ISEPFH | Output | Selected entry points |
| PAC7IE | | Report | Validation report |

Return Codes :

- 0 : OK.
- 12 : System error

Removal of duplicate entry points: PAN215

| Code | Physical name | Type | Label |
|--------|-----------------------------|--------|------------------------------|
| PAC7FH | &&ISEPFH | Input | Selected entry points |
| PAC7HF | &INDUN..&USER..&BASE.FH(+1) | Output | Sorted selected entry points |

| Code | Physical name | Type | Label |
|----------|---------------------------|--------|-----------------------------------|
| PAC7FR | &INDUN.&USER.&BASE.FR(+1) | Output | Reduced entry points to be purged |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

.Return codes:

- 0 : OK.
- 12 : System error.

ISEP - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - IMPACT ANALYSIS : SELECTION OF ENTRY POINTS -
/**
/** -----
/**
/** THE ISEP PROCEDURE IS DESIGNED TO SELECT THE ENTRY
/** POINTS -- DATA ELEMENTS AND/OR CHARACTER STRINGS --
/** WHICH WILL BE USED AS CRITERIA BY THE IMPACT
/** ANALYSIS (IANA PROCEDURE).
/**
/** -----
/**
/**BVP ISEP  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**      USER=,                               PACKAGE CODE FOR IMPACT ANALYSIS
/**      INDSV='$INDSV',                       INDEX OF SYSTEM VSAM FILES
/**      INDSN='$INDSN',                       INDEX OF SYSTEM NON-VSAM FILES
/**      INDUV='$INDUV',                       INDEX OF USER VSAM FILES
/**      INDUN='$INDUN',                       INDEX OF USER NON VSAM FILES
/***:      VSAMCAT='$VCAT',                     USER VSAM CATALOG
/***:      SYSCAT='$SCAT',                     SYSTEM VSAM CATALOG
/**      STEPLIB='$HLQ..SBVPMBR8',           LIBRARY OF LOAD-MODULES
/**      SORTLIB='$BIBT',                     SORT LIBRARY
/**      LSR='$LSR',                           LSR BATCH SYSTEM NAME
/**      OUT=$OUT,                             OUTPUT CLASS
/**      DSCB='$DSCB',                         DSCB MODEL FILE
/**      SPAFH=(TRK,(30,10)),                 SPACE OF FH FILE
/**      SPAFR=(TRK,(30,10)),                 SPACE OF FR FILE
/**      VOLS='SER=$VOLUN',                   FH AND FO FILES VOLUME
/**      UNITS=$UNITUN,                       FH AND FO FILES UNIT
/**      CYL=5,                               SIZE OF WORK OF SORT
/**      UWK=$UWK                             WORK UNIT
/*******
/**INPUT  EXEC PGM=BVPTU001
/***-----

```

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&ISEPMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN210 EXEC PGM=BVPAN210
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FP DD DSN=&INDUV..&USER..&BASE.FP,DISP=SHR
//PAC7FH DD DSN=&&ISEPFH,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFH,
// DCB=(RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7IE DD SYSOUT=&OUT
//PAC7MB DD DSN=&&ISEPMB,DISP=(OLD,DELETE)
//SYSUDUMP DD SYSOUT=&OUT
//PAN215 EXEC PGM=BVPAN215
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR

```

```

//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7FH DD DSN=&&ISEPFH,DISP=(OLD,DELETE)
//PAC7HF DD DSN=&INDUN..&USER..&BASE.FH(+1),
//      DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      SPACE=&SPAFH,
//      DCB=(&DSCB,RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7FR DD DSN=&INDUN..&USER..&BASE.FR(+1),
//      DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      SPACE=&SPAFR,
//      DCB=(&DSCB,RECFM=FB,BLKSIZE=21600,LRECL=72)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

ISOS - Selection of Strings and Operators

ISOS - Introduction

ISOS is a complement to the ISEP procedure. Its purpose is to select the following items:

- VA Pac-processed dates, such as DATOR and DAT8, that will be used as entry points to perform the impact analysis from the first iteration (IANA procedure),
- Character-strings, without considering them as entry points (such as ORDER BY). For the strings which provide entry points, see the description of the 'S'-type line in the ISEP procedure's USER INPUT section,
- Operators used in procedural code (-P) lines, such as ADT. Some of these operators trigger the generation of date-type entry points (such as DATOR for ADT),
- Lines that use constant values, either defined (VALUE), moved (MOVE), or conditioned ('IF').

The restoration of the entities which use these operators and character-strings can be executed on request (IPIA procedure).

Narrowing the scope of selection

For better performance, it is advisable to narrow the scope of the selection. This can be done at two different levels, and should always be done before running the procedure.

- Via the UXSR procedure, documented in 'The Administrator's Procedures' manual, you can create another Development Database. The new Database is a subset (restructured and/or renamed) of the initial Database. The analysis will be performed on this subset.
- Via the INFP utility, documented in this manual: 'FP File Initialization (Impact Analysis)', you can decide to restrict the scope of the selection to entities of a particular type or types, or to particular entities of a given type. Further selection options are also available.

The selection context's identification line (*-line) is required. It allows you to specify, besides the session, the Library from which you want to build the sub-network that will be analyzed (view Z1).

Three types of selection may be used (see below). At least one type of selection is required, no particular type being requested.

The selection may include more than one type of selection, and more than one command line for each type.

- The 'D'-type line allows you to request the extraction of date-type Elements handled by VisualAge Pacbase.
The maximum number of 'D'-lines is 40.
- The 'C'-type line allows you to extract character-strings that are likely to include one or more blanks. In this case, the separator must be specified, and the number of blanks is significant. These strings are not entry points.
The maximum number of 'C'-lines is 50 characters for each one of the three search domains.
- The 'O'-type line allows you to select operators processed in -P lines.
The maximum number of 'O'-lines is 50.

Execution conditions

None.

Abnormal execution

Whatever the cause of an abnormal ending, the procedure may be restarted as it is after correction of the problem.

Result

The output of the ISOS procedure is:

- an 'FH' file which contains the selected entry points, to be used by the IANA procedure,

- an 'FR' file which contains the entry points to be purged, to be used by the IANA procedure,
- an 'FO' file, which contains the analysis results, to be used by the IANA or IPIA procedure.

ISOS - User Input

Only one '*'-line (required, located at the beginning of the stream):

| Position | Length | Value | Meaning |
|----------|--------|------------|---|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | Password |
| 19 | 3 | bbb | Code of the highest Library in the sub-network |
| 22 | 4 | ssss | Session number (blank if current session) |
| 26 | 1 | | Session status (' ' or 'T') |
| 28 | 1 | 'F' or 'E' | Language code if different from that of the site (bilingual sites only) |
| 69 | 3 | iii | Code of the lowest Library in the sub-network (optional) |

One 'D'-line for the selection of generated dates (optional):

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 2 | 1 | 'D' | Line code |
| 3 | 9 | | Code of generated date Element to be extracted (which must be recognized by the system) |

One 'O'-line for the selection of operators (optional):

| Position | Length | Value | Meaning |
|----------|--------|-------|--|
| 2 | 1 | 'O' | Line code |
| 3 | 3 | | Code of searched operator (which must be recognized by the system) |

One 'C'-line for the selection of character strings (optional):

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 2 | 1 | 'C' | Line code |
| 3 | 1 | | End-of-string separator (Required if the string contains at least one blank) |
| 4 | 31 | | Code of searched string. (Must be ended by the separator if a separator is specified) |
| 35 | 1 | | Where the string is to be searched: |
| | | 'D' | Search in the Definition part (-W of Programs and/or Screens, and -9 of programs) |
| | | 'T' | Search in Procedural Code part (-P of programs and/or screens, -8, -9, -SC of programs, -CE and -CS of screens) |
| | | 'R' | Search in the Report specific part: .Category condition and Structure .Source Element code (Struct.) |
| | | ' ' | Search in the three above mentioned parts |

One 'V'-line for the selection of constant values (optional):

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 2 | 1 | 'V' | Line code |
| 3 | 1 | | Beginning-of-value separator Required (either ' or ") |
| 4 | 31 | | Code of searched value Required, ending with the separator (either ' or ") |
| 35 | 1 | | Where the constant is to be searched |
| | | 'D' | Search in the Definition part (-W of Programs and/or Screens, and -9 of Programs) |
| | | 'T' | Search in the Procedural Code part (-P of Programs and/or Screens, -8, -9, -SC of Programs, -CE and -CS of Screens) |
| | | 'R' | Search in the Report specific part: .Category condition and Structure .Source Element code (Struct.) |
| | | ' ' | Search in the three above mentioned parts |

ISOS - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Selection of strings and operators: PAN212

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|-------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administrator Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administrator Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administrator Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7FP | &INDUV..&USER..&BASE.FP | Input | Entities to be analyzed |
| PAC7MB | &&ISOSMB | Input | User input |
| PAC7FH | &&ISOSFH | Output | Selected entry points (length=160) |
| PAC7MF | &&ISOSFO | Output | Impact analysis result (length=266) |
| PAC7IE | | Report | Validation control |

.Return Codes :

- 0: OK
- 12: System error

Deletion of duplicate entry points: PAN215

| Code | Physical name | Type | Label |
|----------|-----------------------------|--------|-----------------------------------|
| PAC7FH | &&ISOSFH | Input | Selected entry points |
| PAC7HF | &INDUN..&USER..&BASE.FH(+1) | Output | Sorted selected entry points |
| PAC7FR | &INDUN..&USER..&BASE.FR(+1) | Output | Reduced entry points to be purged |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |

| Code | Physical name | Type | Label |
|----------|---------------|------|-------|
| SORTWK03 | | Sort | |

Return codes :

- 0 : OK
- 12 : System error

Update of impact analysis results: PAN260

| Code | Physical name | Type | Label |
|----------|-----------------------------|--------|---|
| PAC7MF | &&ISOSFO | Input | Impact analysis result (for that iteration) |
| PAC7OF | &INDUN..&USER..&BASE.FO(0) | Input | Results from preceding analysis |
| PAC7FO | &INDUN..&USER..&BASE.FO(+1) | Output | Sorted impact-analysis results |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK.
- 12 : System error.

ISOS - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/** - IMPACT ANALYSIS: SELECTION OF STRINGS AND OPERATORS
/**
/** -----
/**
/** ISOS IS A COMPLEMENT TO THE ISEP PROCEDURE.
/** FOR BETTER PERFORMANCE, IT IS ADVISABLE TO NARROW THE
/** SCOPE OF THE SELECTION. THIS CAN BE DONE
/** AT TWO DIFFERENT LEVELS, AND SHOULD ALWAYS
/** BE DONE BEFORE RUNNING THE PROCEDURE.
/**
/** -----
/**
/**BVPILOS  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**          USER=,                          PACKAGE CODE FOR IMPACT ANALYSIS
/**          INDSV='$INDSV',                  INDEX OF SYSTEM VSAM FILES

```



```

//      INDSN='$INDSN',           INDEX OF SYSTEM NON-VSAM FILES
//      INDUV='$INDUV',           INDEX OF USER VSAM FILES
//      INDUN='$INDUN',           INDEX OF USER NON-VSAM FILES
//*:    VSAMCAT='$VCAT',           USER VSAM CATALOG
//*:    SYSTCAT='$SCAT',           SYSTEM VSAM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//      LSR='BLSR',               LSR BATCH SYSTEM NAME
//      SORTLIB='$BIBT',          SORT LIBRARY
//      OUT=$OUT,                 OUTPUT CLASS
//      DSCB='$DSCB',             DSCB MODEL FILE
//      SPAFH=(TRK,(30,10)),      SPACE OF CRITERIA FILE
//      SPAFR=(TRK,(30,10)),      SPACE OF CRITERIA FILE
//      SPAFO=(TRK,(30,10)),      SPACE OF RESULTS FILE
//      VOLS='SER=$VOLUN',        FH AND FO FILES VOLUME
//      UNITS=$UNITUN,            FH AND FO FILES UNIT
//      CYL=10,                   'SPACE' OF SORT FILE
//      UWK=$UWK                   WORK UNIT
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&ISOSMB,DISP=(,PASS),
//        UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//        DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//        DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//        DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//        DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN212 EXEC PGM=BVPAN212
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE  DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN  DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR  DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR   DD DSN=&INDSV..BVPGN,DISP=SHR

```

```

//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNSLR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FP DD DSN=&INDUV..&USER..&BASE.FP,DISP=SHR
//PAC7FH DD DSN=&&ISOSFH,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFH,
// DCB=(RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7MF DD DSN=&&ISOSFO,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFO,
// DCB=(RECFM=FB,BLKSIZE=26600,LRECL=266)
//PAC7IE DD SYSOUT=&OUT
//PAC7MB DD DSN=&&ISOSMB,DISP=(OLD,DELETE)
//SYSUDUMP DD SYSOUT=&OUT
//PAN215 EXEC PGM=BVPAN215
//*-----
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7FH DD DSN=&&ISOSFH,DISP=(OLD,DELETE)
//PAC7HF DD DSN=&INDUN..&USER..&BASE.FH(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAFH,
// DCB=(&DSCB,RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7FR DD DSN=&INDUN..&USER..&BASE.FR(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAFR,
// DCB=(&DSCB,RECFM=FB,BLKSIZE=21600,LRECL=72)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PAN260 EXEC PGM=BVPAN260
//*-----
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT

```

```

//SYSPRINT DD SYSOUT=&OUT
//PAC70F DD DSN=&INDUN..&USER..&BASE.FO(0),DISP=SHR
//PAC7MF DD DSN=&&ISOSFO,DISP=(OLD,DELETE)
//PAC7FO DD DSN=&INDUN..&USER..&BASE.FO(+1),
//      DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      SPACE=&SPAFO,
//      DCB=(&DSCB,RECFM=FB,BLKSIZE=26600,LRECL=266)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

IMFH - Merge of FH Files - Creation of FH and FR

IMFH - Introduction

The IMFH procedure allows you to merge two or more FH files (selected entry points) so as to:

- Have only one FH file, after eliminating possible duplicates;
- Obtain an FR file (entry points to be purged) in phase with the FH file created.

This procedure should be used when you want to merge the FH file produced by the ISEP procedure with that issued by the ISOS procedure.

A subsidiary use of this procedure is to recreate the FR file from an FH file.

Result

The ISEP procedure outputs two files which are to be used by the IANA procedure:

- an 'FH' file which contains the selected entry points,
- an 'FR' file which contains the entry points to be purged.

IMFH - Description of Steps

Deletion of duplicate entry points: PAN215

| Code | Physical name | Type | Label |
|----------|---|--------|------------------------------------|
| PAC7FH | &INDUN..&USER..&BASE.FH(0) &INDUN..&USER..&BASE.FH(-1) | Input | Selected entry points to be merged |
| PAC7HF | &INDUN..&USER..&BASE.FH(+1) | Output | Sorted selected entry points |
| PAC7FR | &INDUN..&USER..&BASE.FR(+1) | Output | Reduced entry points to be purged |
| SORTWK01 | | Sort | |

| Code | Physical name | Type | Label |
|----------|---------------|------|-------|
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK.
- 12 : System error.

IMFH - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/* - IMPACT ANALYSIS: MERGE FH FILES AND CREATION FR FILE
/*
/* -----
/*
/* THIS PROCEDURE SHOULD BE USED WHEN YOU WANT TO MERGE
/* THE FH FILE PRODUCED BY THE ISEP PROCEDURE WITH THAT
/* ISSUED BY THE ISOS PROCEDURE.
/*
/*
/* -----
/*
//BVPIMFH  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          USER=,                        PACKAGE CODE FOR IMPACT ANALYSIS
//          INDUN='$INDUN',                INDEX OF USER NON-VSAM FILES
//*:       VSAMCAT='$VCAT',                USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',              SORT LIBRARY
//          OUT=$OUT,                      OUTPUT CLASS
//          DSCB='$DSCB',                  DSCB MODEL FILE
//          SPAFH='(TRK,(30,10))',        SPACE OF CRITERIA FILE
//          SPAFR='(TRK,(30,10))',        SPACE OF CRITERIA FILE
//          VOLS='SER=$VOLUN',             FH AND FO FILES VOLUME
//          UNITS=$UNITUN,                 FH AND FO FILES UNIT
//          CYL=10,                         'SPACE' OF SORT FILE
//          UWK=$UWK                        WORK UNIT
//*****
//PAN215 EXEC PGM=BVPAN215
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7FH DD DSN=&INDUN..&USER..&BASE.FH(0),DISP=SHR

```

```

//      DD DSN=&INDUN..&USER..&BASE.FH(-1),DISP=SHR
//PAC7HF DD DSN=&INDUN..&USER..&BASE.FH(+1),
//      DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      SPACE=&SPAFH,
//      DCB=(&DSCB,RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7FR DD DSN=&INDUN..&USER..&BASE.FR(+1),
//      DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      SPACE=&SPAFR,
//      DCB=(&DSCB,RECFM=FB,BLKSIZE=21600,LRECL=72)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

INFQ - FQ File Reinitialization (Impact Analysis)

INFQ - Introduction

The INFQ procedure reinitializes the FQ file, which accumulates all the search criteria that have already been impacted by the analysis. This accumulation prevents these criteria from being analyzed again in future analyses.

This action should be performed before a new impact analysis either because the entry points have changed or because the analysis context has changed.

However, it must not be used between two iterations of the same impact analysis.

User input

A '*' line with the user code and password.

Result

The procedure outputs a reinitialized file of search criteria (FQ).

INFQ - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

FQ file Reinitialization : PAN200

| Code | Physical name | Type | Label |
|--------|---------------|-------|----------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |

| Code | Physical name | Type | Label |
|--------|-----------------------------|--------|---|
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7MB | &&INFQMB | Input | User input |
| PAC7FQ | &INDUN..&USER..&BASE.FQ(+1) | Output | Impacted criteria reinitialized sequential file |
| PAC7DD | | Report | Error report |

INFQ - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - IMPACT ANALYSIS:  INITIALIZATION OF "FQ" FILE -
/**
/** -----
/**
/** THIS ACTION SHOULD BE PERFORMED BEFORE A NEW IMPACT
/** ANALYSIS EITHER BECAUSE THE ENTRY POINTS HAVE CHANGED
/** OR BECAUSE THE ANALYSIS CONTEXT HAS CHANGED.
/**
/** -----
/**
//BVPINFQ  PROC BASE=$BASE,           CODE OF DEVPT DATABASE
//          USER=,                   PACKAGE CODE FOR IMPACT ANALYSIS
//          INDSV='$INDSV',           INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',           INDEX OF SYSTEM NON-VSAM FILES
//          INDUV='$INDUV',           INDEX OF USER VSAM FILES
//          INDUN='$INDUN',           INDEX OF USER NON-VSAM FILES
//*:       VSAMCAT='$VCAT',           USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',           SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          LSR='BLSR',               LSR BATCH SYSTEM NAME
//          UWK=$UWK,                 WORK UNIT
//          OUT=$OUT,                 OUTPUT CLASS
//          DSCB='$DSCB',             DSCB MODEL FILE
//          VOLS='SER=$VOLUN',        RESULTS FILE VOLUME
//          UNITS=$UNITUN            RESULTS FILE UNIT
//*****
//INPUT   EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR

```

```

//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&INFQMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN200 EXEC PGM=BVPAN200
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FQ DD DSN=&INDUN..&USER..&BASE.FQ(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=(TRK,1),
// DCB=(&DSCB,RECFM=FB,LRECL=100,BLKSIZE=21600)
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&INFQMB,DISP=(OLD,DELETE)

//SYSUDUMP DD SYSOUT=&OUT

```

IGRA - Breaking down of Group Fields

IGRA - Introduction

The IGRA procedure breaks down group fields into Elementary Fields. These group fields can be:

- Entry points detected by the ISEP procedure.
- Impact search criteria obtained by running the IANA procedure.

The IGRA procedure is optional and does not generate any impact search criterion.

Before running the IGRA procedure, you may purge:

- Entry points --after executing the ISEP procedure.
- Impact search criteria --after executing the IANA procedure which precedes.

In both cases, deletions are made in the FR file (via an editor) by inhibiting them (value 'E' in the action code of the corresponding lines), in order to save them for future executions of IANA.

It is not necessary to eliminate non-Group fields since they will simply be ignored by the procedure.

The notions of 'level' and 'iterations' are not relevant for the IGRA procedure.

Entry points (first iteration) or impact search criteria (further iterations) are printed once the purged criteria have been taken into account. This printout sorts criteria into 'accepted' and 'rejected' criteria.

The impact results file may either be empty or contain the results of other IANA, ISOS, or IGRA executions, either in the same execution context or in different contexts. This allows you to compound the results of all iterations of the impact analysis for one or several contexts.

Restitution of all the information for a given context may be customized (parameter setting) when printing with the IPIA procedure.

The file of Entities to be analyzed (FP) is used as input to this procedure. It contains a list of Entities or Entity Types which should be analyzed. If no user input is entered in this file before its initialization by the INFP procedure, all analyzable Entities will be analyzed.

Entities to be analyzed are specified as follows: 3-character Type, and 6-character code (***** being the Entity generic code).

Execution conditions

None, except that the FH file (entry points or impact search criteria) must exist and must not be empty.

Abnormal execution

Whatever the cause of the abnormal ending, the procedure may be restarted as it is after correcting the problem. However, the status of generation files (FH, FR, and FO) should be checked.

Result

The procedure outputs a file which contains the analysis results (FO) to be used in the IPIA procedure.

User input

One '*' line with user code and password.

IGRA - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Recognition of purged criteria: PAN230

| Code | Physical name | Type | Label |
|----------|----------------------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7MB | &&IGRAMB | Input | User input |
| PAC7FH | &INDUN..&USER..&BASE.FH(0) | Input | Search criteria file |
| PAC7FR | &INDUN..&USER..&BASE.FR(0) | Input | Reduced file of purged criteria |
| PAC7HF | &&IGRAHF | Output | Search criteria file (length=160) |
| PAC7DD | | Report | Error file |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes :

- 0 : OK
- 12 : System error

Printing of entry points: PAN220

| Code | Physical name | Type | Label |
|----------|----------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PAC7HF | &&IGRAHF | Input | Sorted criteria file |
| PAC7IL | | Report | List of accepted/rejected criteria |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes :

- 0 : OK
- 12 : System error

Breaking down of group fields: PAN255

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Data index file |
| PAC7FP | &INDUV..&USER..&BASE.FP | Input | Entities to analyze |
| PAC7FH | &&IGRAHF | Input | Impacted criteria |
| PAC7MF | &&IGRAFO | Output | Impact analysis results (length=266) |

Return Codes :

- 0 : OK

- 12 : System error

Update of impact analysis results: PAN260

| Code | Physical name | Type | Label |
|----------|-----------------------------|--------|---------------------------------------|
| PAC7MF | &&IGRAFO | Input | Impact analysis result (by level) |
| PAC7OF | &INDUN..&USER..&BASE.FO(0) | Input | Results of previous analysis |
| PAC7FO | &INDUN..&USER..&BASE.FO(+1) | Output | Sorted results of the impact analysis |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK
- 12 : System error

IGRA - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - IMPACT ANALYSIS:  PRINT OF "FQ" FILE  -
/**
/** -----
/**
/** THE IGRA PROCEDURE BREAKS DOWN GROUP FIELDS
/** INTO ELEMENTARY FIELDS:
/** 1. ENTRY POINTS DETECTED BY THE ISEP PROCEDURE, IF
/** THEY ARE OF THE GROUP TYPE.
/** 2. IMPACT SEARCH CRITERIA OBTAINED BY RUNNING THE IANA
/** PROCEDURE, IF THEY ARE OF THE GROUP TYPE.
/**
/** -----
/**
//BVPIGRA  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          USER=,                          PACKAGE CODE FOR IMPACT ANALYSIS
//          INDSV='$INDSV',                  INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',                  INDEX OF SYSTEM NON-VSAM FILES
//          INDUV='$INDUV',                  INDEX OF USER VSAM FILES
//          INDUN='$INDUN',                  INDEX OF USER NON-VSAM FILES
/**:      VSAMCAT='$VCAT',                    USER VSAM CATALOG
/**:      SYSCAT='$SCAT',                     SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',        LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',                  SORT LIBRARY
//          OUT=$OUT,                          OUTPUT CLASS
//          DSCB='$DSCB',                      DSCB MODEL FILE

```

```

//          LSR='BLSR',                LSR BATCH SYSTEM NAME
//          SPAFO='(TRK,(30,10))',     SPACE OF RESULTS FILE
//          SPAFH='(TRK,(30,10))',     SPACE OF CRITERIA FILE
//          VOLS='SER=$VOLUN',         RESULTS FILE VOLUME
//          UNITS=$UNITUN,             RESULTS FILE UNIT
//          CYL=5,                     SIZE OF WORK FILE
//          UWK=$UWK                   WORK UNIT
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&IGRAMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN230 EXEC PGM=BVPAN230
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DD DD SYSOUT=&OUT
//PAC7FH DD DSN=&INDUN..&USER..&BASE.FH(0),DISP=SHR
//PAC7FR DD DSN=&INDUN..&USER..&BASE.FR(0),DISP=SHR
//PAC7HF DD DSN=&&IGRAHF,
//          DISP=(,PASS,DELETE),
//          UNIT=&UWK,

```

```

//          SPACE=&SPAFH,
//          DCB=(RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7MB DD DSN=&&IGRAMB,DISP=(OLD,DELETE)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAN220 EXEC PGM=BVPAN220,COND=(0,NE,PAN230)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7HF DD DSN=&&IGRAHF,DISP=(OLD,PASS)
//PAC7IL DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PAN255 EXEC PGM=BVPAN255,COND=(0,NE,PAN230)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FH DD DSN=&&IGRAHF,DISP=(OLD,DELETE)
//PAC7FP DD DSN=&INDUV..&USER..&BASE.FP,DISP=SHR
//PAC7MF DD DSN=&&IGRAFO,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFO,
// DCB=(RECFM=FB,BLKSIZE=26600,LRECL=266)
//SYSUDUMP DD SYSOUT=&OUT
//PAN260 EXEC PGM=BVPAN260,COND=(0,NE,PAN230)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR

```

```

//          DD DSN=$BCOB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC70F   DD DSN=&INDUN..&USER..&BASE.FO(0),DISP=SHR
//PAC7MF   DD DSN=&&IGRAFO,DISP=(OLD,DELETE)
//PAC7FO   DD DSN=&INDUN..&USER..&BASE.FO(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAFO,
//          DCB=(&DSCB,RECFM=FB,BLKSIZE=26600,LRECL=266)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

IANA - Impact Search Criteria

IANA - Introduction

The IANA procedure is used to search impacted Data Elements and character-strings according to:

- The entry points provided by the ISEP procedure when IANA is run for the first time,
- The impact search criteria produced by a preceding execution of IANA.

IANA is therefore an iterative process, which runs until no more impact search criteria are found.

Prior to an IANA execution, you can inhibit:

- Entry points, after the execution of the ISEP procedure,
- Impact search criteria, after a preceding execution of the IANA procedure.

In both cases, deletions are made in the FR file, (under an editor) either by physical deletion, or by inhibition (value 'E' in the action code of the corresponding lines).

The entry points (first iteration) or impact search criteria (further iterations) are printed once the purged criteria have been taken into account. This printout sorts criteria into 'accepted' and 'rejected' criteria. The file which contains the already impacted criteria (FQ) may be reinitialized if you do not need to save them.

However, it is recommended to reinitialize this file before the first execution of IANA which follows a new execution of ISEP. To reinitialize the FQ file, run the INFQ procedure.

The impact analysis file may either be empty or contain the results of different execution contexts. It allows to compound the results of all iterations of the impact analysis for a given context.

The FP file used as input for the analysis procedures, contains the list of the entities or entity types to be analyzed. If no user input is entered in this file before it is initialized by the INFP procedure, all analyzable entities will be analyzed.

Entities which are to be analyzed are specified in the FP file via the following coding: type coded on 3 characters, entity coded on 6 characters (***** being the generic entity code).

Execution conditions

The FH file -- entry points or impact search criteria -- must exist and must not be empty.

Abnormal execution

Whatever the cause of the abend, you can run the procedure again as it is, after the problem has been solved.

However, the status of the FH, FR, and FO generation files should be checked.

User input

One '*' line with user code and password.

This procedure is iterative as long as the FH file (impact search criteria) is not empty (return code set to value 4 if empty, and to value 0 otherwise).

Result

This procedure outputs a file which contains the analysis results (FO) to be used in the IPIA procedure.

IANA - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Initialization of the KSDS work file: IDCAMS

Indexation of sequential file FQ: IDCAMS

| Code | Physical name | Type | Label |
|------|----------------------------|--------|---|
| IN1 | &INDUN..&USER..&BASE.FQ(0) | Input | Previously-processed search criteria (sequential) |
| OU1 | &INDUV..SYIANA.&WORK | Output | Previously-processed search criteria (indexed) |

Recognition of criteria after the purge: PAN230

| Code | Physical name | Type | Label |
|----------|----------------------------|--------|--|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7MB | &&IANAMB | Input | User input |
| PAC7FH | &INDUN..&USER..&BASE.FH(0) | Input | Search criteria file |
| PAC7FR | &INDUN..&USER..&BASE.FR(0) | Input | Search criteria after purge (reduced file) |
| PAC7HF | &&IANA HF | Output | Search criteria file (length=160) |
| PAC7DD | | Report | Error report |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes :

- 0 : OK
- 12 : System error

Printing of entry points: PAN220

| Code | Physical name | Type | Label |
|--------|----------------|-------|-----------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PAC7HF | &&IANA HF | Input | Sorted criteria |

| Code | Physical name | Type | Label |
|----------|---------------|--------|--------------------------------------|
| PAC7IL | | Report | List of accepted / rejected criteria |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes :

- 0 : OK
- 12 : System error

Impact analysis: PAN250

| Code | Physical name | Type | Label |
|--------|-------------------------|----------------|--|
| PAC7AE | &INDSV..BVPAE | Input | Error messages |
| PACGGN | &INDSV..BVPGN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVPGR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVPGU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7AY | &INDUV..&BASE.AY | Input | Development Database extension data |
| PAC7FP | &INDUV..&USER..&BASE.FP | Input | File of entities to be analyzed |
| PAC7FH | &&IANAHF | Input | Impacted criteria |
| PAC7FQ | &INDUV..SYIANA.&WORK | Input / Output | Impacted criteria already processed |
| PAC7HF | &&IANAFH | Output | New impacted criteria (length = 160) |
| PAC7MF | &&IANAFO | Output | Impact analysis results (length = 266) |

Return codes:

- 0 : OK

- 4 : OK. Iteration ended.
- 12 : System error

Update of impact analysis results: PAN260

| Code | Physical name | Type | Label |
|----------|-----------------------------|--------|-----------------------------------|
| PAC7MF | &&IANAF0 | Input | Impact analysis results (level) |
| PAC7OF | &INDUN..&USER..&BASE.FO(0) | Input | Results of previous analysis |
| PAC7FO | &INDUN..&USER..&BASE.FO(+1) | Output | Sorted results of impact analysis |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK
- 12 : System error

Removal of duplicate entry points: PAN215

| Code | Physical name | Type | Label |
|----------|-----------------------------|--------|-----------------------------------|
| PAC7FH | &&IANAFH | Input | Selected entry points |
| PAC7HF | &INDUN..&USER..&BASE.FH(+1) | Output | Sorted selected entry points |
| PAC7FR | &INDUN..&USER..&BASE.FR(+1) | Output | Reduced entry points to be purged |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK
- 12 : System error

Transforming the FQ VSAM file into a sequential file: IDCAMS

| Code | Physical name | Type | Label |
|------|----------------------|-------|--|
| IN1 | &INDUV..SYIANA.&WORK | Input | Previously processed search criteria (indexed) |

| Code | Physical name | Type | Label |
|------|--------------------------------|--------|---|
| OU1 | &INDUN.&USER.. &BASE.FQ(+1) | Output | Previously processed search criteria (sequential) |

DELETE the impacted criteria already processed: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------|
| SYSIN | &&DLSYIANA | Input | |

IANA - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**              - IMPACT ANALYSIS -
/**
/** -----
/**
/** THE IANA PROCEDURE IS USED TO SEARCH DATA ELEMENTS AND
/** CHARACTER-STRINGS ACCORDING TO:
/** 1.THE ENTRY POINTS PROVIDED BY THE ISEP PROCEDURE WHEN
/**    IANA IS RUN FOR THE FIRST TIME,
/** 2.THE IMPACT SEARCH CRITERIA PRODUCED
/**    BY A PRECEDING EXECUTION OF IANA.
/** IANA IS THEREFORE AN ITERATIVE PROCESS, WHICH RUNS
/** UNTIL NO MORE IMPACT SEARCH CRITERIA ARE FOUND.
/**
/** -----
/**
/**BVPIANA  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**          USER=,                          PACKAGE CODE FOR IMPACT ANALYSIS
/**          INDSV='$INDSV',                  INDEX OF SYSTEM VSAM FILES
/**          INDSN='$INDSN',                  INDEX OF SYSTEM NON-VSAM FILES
/**          INDUV='$INDUV',                  INDEX OF USER VSAM FILES
/**          INDUN='$INDUN',                  INDEX OF USER NON-VSAM FILES
/***:      VSAMCAT='$VCAT',                    USER VSAM CATALOG
/***:      SYSCAT='$SCAT',                     SYSTEM VSAM CATALOG
/**          STEPLIB='$HLQ..SBVPMBR8',        LIBRARY OF LOAD-MODULES
/**          SORTLIB='$BIBT',                  SORT LIBRARY
/**          OUT=$OUT,                          OUTPUT CLASS
/**          DSCB='$DSCB',                      DSCB MODEL FILE
/**          LSR='BLSR',                         LSR BATCH SYSTEM NAME
/**          SPAFO='(TRK,(30,10))',            SPACE OF RESULTS FILE
/**          SPAFH='(TRK,(30,10))',            SPACE OF CRITERIA FILE
/**          SPAFQ='(TRK,(30,10))',            SPACE OF FQ FILE
/**          WORK='TEST',                       SUFFIX OF FQ WORK FILE
/**          VOLS='SER=$VOLUN',                 RESULTS FILE VOLUME
/**          UNITS=$UNITUN,                     RESULTS FILE UNIT
/**          CYL=5,                             SIZE OF WORK OF SORT
/**          UWK=$UWK                             WORK UNIT

```

```

//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&IANAMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&WORK'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYIANA),DISP=SHR
//PACROU DD DSN=&&DFSYIANA,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYIANA,DISP=(OLD,DELETE)
//STEP1 EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//IN1 DD DSN=&INDUN..&USER..&BASE.FQ(0),DISP=SHR
//OU1 DD DSN=&INDUV..SYIANA.&WORK,DISP=OLD
//SYSIN DD DSN=&INDSN..BVPSY(REPRO),DISP=SHR
//*
//PAN230 EXEC PGM=BVPAN230
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR

```

```

//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DD DD SYSOUT=&OUT
//PAC7FH DD DSN=&INDUN..&USER..&BASE.FH(0),DISP=SHR
//PAC7FR DD DSN=&INDUN..&USER..&BASE.FR(0),DISP=SHR
//PAC7HF DD DSN=&&IANAHF,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFH,
// DCB=(RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7MB DD DSN=&&IANAMB,DISP=(OLD,DELETE)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAN220 EXEC PGM=BVPAN220,COND=(0,NE,PAN230)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7HF DD DSN=&&IANAHF,DISP=(OLD,PASS)
//PAC7IL DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PAN250 EXEC PGM=BVPAN250,COND=(0,NE,PAN230)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')

```

```

//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FH DD DSN=&IANAHF,DISP=(OLD,DELETE)
//PAC7FQ DD DSN=&INDUV..SYIANA.&WORK,DISP=OLD
//PAC7FP DD DSN=&INDUV..&USER..&BASE.FP,DISP=SHR
//PAC7HF DD DSN=&IANAFH,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFH,
// DCB=(RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7MF DD DSN=&IANAFO,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFO,
// DCB=(RECFM=FB,BLKSIZE=26600,LRECL=266)
//SYSUDUMP DD SYSOUT=&OUT
//PAN260 EXEC PGM=BVPAN260,COND=(0,NE,PAN230)
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC70F DD DSN=&INDUN..&USER..&BASE.FO(0),DISP=SHR
//PAC7MF DD DSN=&IANAFO,DISP=(OLD,DELETE)
//PAC7FO DD DSN=&INDUN..&USER..&BASE.FO(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAFO,
// DCB=(&DSCB,RECFM=FB,BLKSIZE=26600,LRECL=266)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PAN215 EXEC PGM=BVPAN215,COND=(0,NE,PAN230)
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7FH DD DSN=&IANAFH,DISP=(OLD,DELETE)
//PAC7HF DD DSN=&INDUN..&USER..&BASE.FH(+1),
// DISP=(,CATLG,DELETE),

```

```

//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAFH,
//          DCB=(&DSCB,RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7FR DD DSN=&INDUN..&USER..&BASE.FR(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAFH,
//          DCB=(&DSCB,RECFM=FB,BLKSIZE=21600,LRECL=72)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//*
//STEP2 EXEC PGM=IDCAMS,COND=(0,NE,PAN230)
//*-----
/*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//IN1 DD DSN=&INDUV..SYIANA.&WORK,DISP=OLD
//OU1 DD DSN=&INDUN..&USER..&BASE.FQ(+1),
//      DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      SPACE=&SPAFQ,
//      DCB=(&DSCB,RECFM=FB,LRECL=100,BLKSIZE=21600)
//*
//SYSIN DD DSN=&INDSN..BVPSY(REPRO),DISP=SHR
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&WORK',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYIANA),DISP=SHR
//PACROU DD DSN=&&DLSYIANA,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYIANA,DISP=(OLD,DELETE)

```

IPFQ - FQ File Printout (Impact Analysis)

IPFQ - Introduction

The IPFQ procedure prints all the entry points and impact search criteria accepted or rejected during a comprehensive impact analysis.

All the criteria and entry points are stored in the FQ file.

IPFQ offers four types of printouts:

- List of accepted entry points
- List of rejected entry points
- List of accepted impact search criteria
- List of rejected impact search criteria.

The printout shows criteria and entry points sorted by alphabetical order within each category, and by definition Library of these criteria.

The printing order for the categories are:

- Character strings
- Element defined in the Dictionary,
- Element defined in Segment Descriptions,
- Element defined in Report Structures,
- Element defined in Screen- or Program-Working sections.

The IPFQ procedure can be used to select the entry points and impact search criteria of one or more categories.

In case of selection, only the selected criteria are printed.

Execution conditions

None, but the FQ file must exist.

Abnormal execution

Whatever the cause of the abnormal ending, the procedure can be restarted as it is, after the problem has been corrected.

Result

The procedure prints the entry points and the search criteria.

IPFQ - User Input

A '*' line with the user code and password.

One 'S' line per criteria selection (optional).

| Position | Length | Value | Meaning |
|----------|--------|-------|-----------------------------------|
| 2 | 1 | 'S' | Line code |
| 3 | 1 | | Type of criterion |
| | | 'E' | Element defined in the Dictionary |
| | | 'C' | Character string |

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| | | 'X' | Group-type Element or Element not defined |
| | | '*' | All types of criteria |
| 4 | 1 | | Source code |
| | | '3' | Line from Segment's -CE |
| | | '6' | Line from Report's -CE |
| | | '7' | -W line of a Screen or Program |
| | | '*' | All sources |
| 6 | 1 | | For the type of field |
| | | 'G' | For a Group field |
| | | ' ' | For an elementary field |
| | | '*' | For all types of fields |

IPFQ - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Extraction of criteria: PAN240

| Code | Physical name | Type | Label |
|--------|----------------------------|--------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7FQ | &INDUN..&USER..&BASE.FQ(0) | Input | Criteria impacted during analysis |
| PAC7MB | &&IPFQMB | Input | User input |
| PAC7FH | &&IPFQFH | Output | Search criteria file |
| PAC7IX | | Report | Output report |

Printing of impacted criteria: PAN220

| Code | Physical name | Type | Label |
|----------|----------------|--------|-----------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PAC7HF | &&IPFQFH | Input | Sorted entry points or criteria |
| PAC7IL | | Report | List of entry points and criteria |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return codes:

- 0 : OK.
- 12 : System error.

IPFQ - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - IMPACT ANALYSIS:  GROUP FIELDS ANALYSIS -
/**
/** -----
/**
/** THE IPFQ PROCEDURE PRINTS ALL THE ENTRY POINTS AND
/** IMPACT SEARCH CRITERIA USED (ACCEPTED OR REJECTED)
/** DURING A THOROUGH IMPACT ANALYSIS.
/** ALL THE CRITERIA AND ENTRY POINTS ARE STORED IN THE FQ
/** FILE.
/** PROCEDURE, IF THEY ARE OF THE GROUP TYPE.
/** -----
/**
/**BVPFQ  PROC BASE=$BASE,           CODE OF DEVPT DATABASE
/**      USER=,                     PACKAGE CODE FOR IMPACT ANALYSIS
/**      INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
/**      INDSN='$INDSN',             INDEX OF SYSTEM NON-VSAM FILES
/**      INDUV='$INDUV',             INDEX OF USER VSAM FILES
/**      INDUN='$INDUN',             INDEX OF USER NON-VSAM FILES
/***:    VSAMCAT='$VCAT',            USER VSAM CATALOG
/***:    SYSTCAT='$SCAT',            SYSTEM VSAM CATALOG
/**      STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
/**      SORTLIB='$BIBT',            SORT LIBRARY
/**      LSR='BLSR',                 LSR BATCH SYSTEM NAME
/**      OUT=$OUT,                   OUTPUT CLASS
/**      SPAFH='(TRK,(30,10))',     SPACE OF CRITERIA FILE
/**      CYL=5,                       SIZE OF SORT FILE
/**      UWK=$UWK                     WORK UNIT
/*******
/**INPUT  EXEC PGM=BVPTU001
/**-----

```

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&IPFQMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN240 EXEC PGM=BVPAN240
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FH DD DSN=&&IPFQFH,
// DISP=(,PASS,DELETE),
// UNIT=&UWK,
// SPACE=&SPAFH,
// DCB=(RECFM=FB,BLKSIZE=24000,LRECL=160)
//PAC7FQ DD DSN=&INDUN..&USER..&BASE.FQ(0),DISP=SHR
//PAC7IX DD SYSOUT=&OUT
//PAC7MB DD DSN=&&IPFQMB,DISP=(OLD,DELETE)
//PAN220 EXEC PGM=BVPAN220
//*-----
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR

```

```

//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7HF DD DSN=&&IPFQFH,DISP=(OLD,PASS)
//PAC7IL DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

IPEP - Entry Points Printout

IPEP - Introduction

The IPEP procedure produces two types of printouts.

- List of entry points:

This list is obtained after the ISEP procedure, since this procedure selects the entry points.

- List of impact search criteria:

This list is obtained after the IANA procedure, since this procedure selects the impact search criteria.

In the printout, the criteria or entry points are sorted by alphabetical order (Elements and character strings altogether) for each definition library of these criteria.

The order of printing of the categories is:

- character string
- Element defined in Dictionary
- Element defined in Segment Description
- Element defined in Report Structure
- Element defined in the Screen or Program Working Section.

Execution conditions

None, but the FH file must exist.

Abnormal execution

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

Printouts

Printout of entry points.

User input

No user input is required for the execution of the IPEP procedure.

IPEP - Description of Steps

Printing of entry points: PAN220

| Code | Physical name | Type | Label |
|----------|--------------------------|--------|--------------------------|
| PAC7AE | &INDSV.BVPAE | Input | Error messages |
| PAC7HF | &INDUN.&USER.&BASE.FH(0) | Input | Sorted entry points file |
| PAC7IL | | Report | List of entry points |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

.Return Codes :

- 0 : OK.
- 12 : System error

IPEP - Execution JCL

```
/** -----  
/**      VISUALAGE PACBASE  
/**  
/** -----  
/**      - IMPACT ANALYSIS:  PRINTING OF ENTRY POINTS -  
/**  
/** -----  
/**  
/** THE IPEP PROCEDURE PRODUCES TWO TYPES OF PRINTOUTS.  
/**  1. LIST OF ENTRY POINTS:  
/** THIS LIST IS OBTAINED AFTER THE ISEP PROCEDURE, SINCE  
/** THIS PROCEDURE SELECTS THE ENTRY POINTS.  
/**  2. LIST OF IMPACT SEARCH CRITERIA:  
/** THIS LIST IS OBTAINED AFTER THE IANA PROCEDURE, SINCE  
/** THIS PROCEDURE SELECTS THE IMPACT SEARCH CRITERIA.  
/** -----  
/**  
/**BVIPEP  PROC BASE=$BASE,                CODE OF DEVPT DATABASE  
/**      USER=,                            PACKAGE CODE FOR IMPACT ANALYSIS  
/**      INDSV='$INDSV',                    INDEX OF SYSTEM VSAM FILES  
/**      INDUN='$INDUN',                    INDEX OF USER NON-VSAM FILES  
/***:    VSAMCAT='$VCAT',                  USER VSAM CATALOG  
/***:    SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG  
/**      STEPLIB='$HLQ..SBVPMBR8',        LIBRARY OF LOAD-MODULES  
/**      SORTLIB='$BIBT',                  SORT LIBRARY
```

```

//          OUT=$OUT,                OUTPUT CLASS
//          CYL=5,                    SIZE OF SORTWORK
//          UWK=$UWK                  WORK UNIT
//*****
//PAN220 EXEC PGM=BVPAN220
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7HF DD DSN=&INDUN..&USER..&BASE.FH(0),DISP=SHR
//PAC7IL DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

IPIA - Printing of the Impact Analysis Results

IPIA - Introduction

The IPIA procedure is used to print Reports on the analysis results and to format these results in batch update transactions.

IPIA can output the following reports:

- Analysis results by entry point:
Analysis follow-up of the subsequent iterations. Report requested by value '1' in Position 7 of the P-type user input line.
- List of impact search criteria by entry point:
Valid when the IANA iteration is completed. Report requested by value '1' in Position 8 of the P-type user input line.
- Analysis results by Library:
Results are formatted in batch update transactions (print or file output). Report requested by value '1' in Position 9 of the P-type user input line. Additional option (page and line skips) requested by value '2' in Position 9. File requested by value '1' in Position 12.
- Impacted-instances summary:
List of all impacted instances with the number of impacted lines, for each type of line, not sorted by entry points. Report requested by value '1' in Position 10 of the P-type user input line.

- List of entry points by impacted search criterion for each impacted field: list of the entry points and impact search criteria which originated the impact, after each iteration. Report requested by value '1' in Position 14 of the P-type user input line.
- Statistics:
Number of impacted lines sorted by library and by entity type, all lines considered. Report requested by value '1' in Position 11 of the P-type user input line.
- Character-string analysis:
List of the uses of each character string searched by the ISOS procedure. Report requested by value '1' in Position 19 of the P-type user input line.
- Operator analysis:
List of the uses of each operator searched by the ISOS procedure. Report requested by value '1' in Position 20 of the P-type user input line.
- List of the entities impacted by entry point:
List of the entities impacted by Element-type entry points, all search criteria considered. Report requested by value '1' in Position 21 of the P-type user input line.
- Number of modified lines, dispatched by Description for each entity:
This summary report allows for finer statistics by line types, compounded by Library. Report requested by value '1' in Position 22 of the P-type user input line.
- Constant analysis:
List of uses of each constant searched by the ISOS procedure. Report requested by value '1' in Position 23 of the P-type user input line.

Execution conditions

None, but the FO file must exist and must not be empty.

Abnormal execution

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

Result

The procedure outputs a printout of the analysis results and of the list of transactions sorted by Library.

IPIA - User Input

A line identifying the context (* line) is required. It must be inserted at the beginning of the generated stream.

If you specified a lower library for the ISEP procedure, it must be repeated in this line.

The *-type line must be followed by one P-type, formatted as follows:

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 2 | 1 | 'P' | Line code |
| 3 | 1 | | NOTHING TO ENTER,EXCEPT FOR DOS/VSE |
| | | 'I' | Default option for all hardware |
| | | 'N' | If CURRENT-DATE = MM/DD/YY |
| 4 | 3 | bbb | Library code (this selection is available with requests entered in Positions 9 and 10 only) |
| 7 | 1 | '' | No Result of impact analysis by entry point |
| | | '1' | Result of impact analysis by entry point |
| 8 | 1 | '' | No List of impacted criteria by entry point |
| | | '1' | List of impacted criteria by entry point |
| 9 | 1 | '' | No Printing of formatted results |
| | | '1' | Printing of results formatted as batch update transactions, sorted per Library |
| | | '2' | Same list with page and line skips |
| 10 | 1 | '' | No summary of impacted occurrences |
| | | '1' | List of impacted instances |
| 11 | 1 | '' | No statistics, sorted per Library |
| | | '1' | Statistics, sorted per Library |
| 12 | 1 | '' | Identical to values in Pos. 9 but output is a file instead of a print |
| 13 | 1 | '' | No inhibition of the lines indirectly impacted |
| | | '1' | General option: Inhibition of the lines indirectly impacted (e.g. -CD) |
| 14 | 1 | '' | No list of entry points by impact |
| | | '1' | List of entry points by impact search criterion |
| 15 | 2 | nn | Desired level number (IANA iteration) |
| 17 | 2 | pp | Number of lines printed per page |

| Position | Length | Value | Meaning |
|----------|--------|-------|---|
| 19 | 1 | ' ' | No Result of character-string analysis |
| | | '1' | Result of character-string analysis |
| 20 | 1 | ' ' | No Result of operator analysis |
| | | '1' | Result of operator analysis |
| 21 | 1 | ' ' | No entities impacted by entry point |
| | | '1' | Impacted entities by entry point |
| 22 | 1 | ' ' | No Number of lines per description |
| | | '1' | Number of lines per description |
| 23 | 1 | ' ' | No result of constants analysis |
| | | '1' | Result of constants analysis |
| 24 | 1 | ' ' | No Result of group fields analysis |
| 24 | 1 | '1' | Result of group fields analysis |
| 25 | 10 | | Selection of generated transactions |
| | | Blank | Selection of all entities |
| | | other | Selection among the following entities (you can select several ones): |
| | | 'B' | Database blocks |
| | | 'E' | Elements |
| | | 'F' | Meta-Entities |
| | | 'O' | Screens, C/S Screens... |
| | | 'P' | Programs |
| | | 'R' | Reports |
| | | 'S' | Segments and Data-Structures |
| | | 'T' | Texts |
| | | 'V' | Documents |
| | | '\$' | User Entities |
| 35 | 1 | ' ' | No Recognition of ISOS transactions |
| | | '1' | Recognition of ISOS transactions |

IPIA - Description of Steps

Input recognition: PTU001

Check of VSAM files: IDCAMS

Printing of impact results: PAN270

| Code | Physical name | Type | Label |
|----------|----------------------------|--------|---|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7FO | &INDUN..&USER..&BASE.FO(0) | Input | Impact results |
| PAC7MB | &&IPIAMB | Input | User input |
| PAC7BM | &&IPIABM | Output | Converted user input |
| PAC7GY | &&IPIAGY | Output | PAF transactions for UPDP (length=310) |
| PAC7MV | &&IPIAMV | Output | Batch transactions for printing (length=80) |
| PAC7IF | | Report | Analysis results |
| SORTWK01 | | Sort | |
| SORTWK02 | | Sort | |
| SORTWK03 | | Sort | |

Return Codes :

- 0 : OK
- 12 : System error

Printing of generated transactions: PAN280

| Code | Physical name | Type | Label |
|--------|----------------|--------|---|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PAC7BM | &&IPIABM | Input | User input |
| PAC7MV | &&IPIAMV | Input | Generated batch transactions |
| PAC7VM | &&IPIAVM | | Selected batch transactions (length=80) |
| PAC7IT | | Report | List of transactions per Library |

Return Codes :

- 0 : OK
- 12 : System error

IPIA - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*      - IMPACT ANALYSIS : PRINTING OF RESULTS -
/*
/* -----
/*
/* THE IPIA PROCEDURE IS USED TO PRINT
/* REPORTS ON THE ANALYSIS RESULTS
/* AND TO FORMAT THESE RESULTS IN
/* BATCH UPDATE TRANSACTIONS.
/* -----
/*
//BVPIPIA  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          USER=,                        PACKAGE CODE FOR IMPACT ANALYSIS
//          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',                INDEX OF SYSTEM NON-VSAM FILES
//          INDUV='$INDUV',                INDEX OF USER VSAM FILES
//          INDUN='$INDUN',                INDEX OF USER NON-VSAM FILES
/**:       VSAMCAT='$VCAT',                USER VSAM CATALOG
/**:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//          LSR='BLSR',                    LSR BATCH SYSTEM NAME
//          SORTLIB='$BIBT',                SORT LIBRARY
//          OUT=$OUT,                       OUTPUT CLASS
//          SPAMB='(TRK,(30,10))',         TRANSACTION SPACE
//          SPAGY='(TRK,(100,10))',        GY FILE SPACE
//          CYL=5,                           SIZE OF SORT FILE
//          UWK=$UWK                          WORK UNIT
//*****
//INPUT  EXEC  PGM=BVPTU001
/*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&IPIAMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC  PGM=IDCAMS
/*-----
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR

```

```

//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PAN270 EXEC PGM=BVPAN270
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7FO DD DSN=&INDUN..&USER..&BASE.FO(0),DISP=SHR
//PAC7IF DD SYSOUT=&OUT
//PAC7MB DD DSN=&&IPIAMB,DISP=(OLD,DELETE)
//PAC7BM DD DSN=&&IPIABM,DISP=(,PASS),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MV DD DSN=&&IPIAMV,DISP=(,PASS),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7GY DD DSN=&&IPIAGY,DISP=(,PASS),UNIT=&UWK,
//          SPACE=&SPAGY,
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PAN280 EXEC PGM=BVPAN280,COND=(0,NE,PAN270)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7IT DD SYSOUT=&OUT
//PAC7BM DD DSN=&&IPIABM,DISP=(OLD,DELETE)
//PAC7MV DD DSN=&&IPIAMV,DISP=(OLD,PASS)
//PAC7VM DD DSN=&&IPIAVM,DISP=(,PASS),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT

```

Chapter 8. Methodology Integrity Check

ADM - SSADM Pacdesign Methodology

SADM - Introduction

This procedure is available to the users who have purchased the SSADM Methodology Pacdesign module.

It checks the validity and consistency of occurrences uploaded (by the user) from the WorkStation to the VA Pacbase Repository.

NOTE: The SSADM Methodology and the procedure's functions exist in the English version only. For information on Pacdesign SSADM entities, consult the online help.

Execution conditions

None.

SADM - User Input

One '*' line for library access:

| Position | Length | Value | Meaning |
|----------|--------|----------|--|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |
| 19 | 3 | bbb | Library code |
| 22 | 4 | nnnn | Session number (blank=current session) |
| 26 | 1 | 'T' | Session status if Test session |
| 37 | 25 | | Reserved IMS: request identifier (cf. IMS BATCH PAF) |

Print request lines:

| Position | Length | Value | Meaning |
|----------|--------|-------|------------------------------|
| 2 | 1 | 'T' | Line code |
| 3 | 1 | | Report code |
| | | 'V' | Validation of SSADM Entities |

| Position | Length | Value | Meaning |
|----------|--------|--------|--|
| | | '1' | Cross-boundaries Data flows within a DFD |
| | | '2' | Operational Masters within a DSD |
| | | '3' | All Entities with their attributes |
| 4 | 6 | eeeeee | Entity code (required for '1' or '2') |

Printed output

This procedure prints the following, based on print requests:

- A 'Validation of SSADM entities' report,
- A 'List of cross-boundaries data flows within a DFD',
- A 'List of operational masters within a DSD',
- A 'List of all entities with their attributes'.

SADM - Description of Steps

Input recognition: PTU001

Preparation of the KSDS work-file DEFINE: PRMSYS

This program is used to parameterize the KSDS work file name. The file DELETE/DEFINE is found in the DFSYSPAF member of the SY parameters PDS. Its Dsname is suffixed with the &USER parameter. The PRMSYS program sets this parameter to the value entered at execution time. The purpose is to allocate different work files from one execution to another and to avoid any conflicts when allocating work files in case of simultaneous executions.

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--------------------------------|
| PACRIN | &INDSN..BVPSY(DFSYSPAF) | Input | File DEFINE skeleton |
| PACROU | &&DFSYSPAF | Output | File DEFINE (SYSIN for IDCAMS) |

Definition of KSDS work file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DFSYSPAF | Input | File DEFINE |

SSADM-entity consistency check: PADM10

| Code | Physical name | Type | Label |
|--------|----------------------|--------------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7MB | &&SADMMB | Input | User input |
| SYSPAF | &INDUV..SYSPAF.&USER | Input/Output | PAF standard KSDS file |
| PAC7EJ | | Report | List of checked SSADM entities |
| PAC7ET | | Report | PAF access report |
| PAC7DD | | Report | List of errors |

Preparation of the KSDS work-file DELETE: PRMSYS

This program's purpose is to parameterize the work file name. The DELETE of the file is found in the DFSYSPAF member of the SY parameters PDS.

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--------------------------------|
| PACRIN | &INDSN..BVPSY(DLSYSPAF) | Input | File DELETE skeleton |
| PACROU | DSN=&&DLSYSPAF | Output | File DELETE (SYSIN for IDCAMS) |

Deletion of the KSDS work-file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DLSYSPAF | Input | File DELETE |

SADM - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - PACDESIGN SSADM INTEGRITY CHECKING -

```

```

/**
/** -----
/**
/** THIS PROCEDURE IS SUPPLIED FOR USERS OF THE WORKSTATION
/** AND THE SSADM PACDESIGN APPLICATION DESIGN METHODOLOGY.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/** COL 2 : "*"
/** COL 3 : USERIDXX
/** COL 11 : PASSWORD
/** COL 19 : (BBB) LIBRARY CODE
/** COL 22 : (4 N) SESSION NUMBER
/** COL 26 : (1 CAR.) SESSION VERSION
/** COL 37 (25 CAR.) RESERVED IMS
/** - COMMAND LINE :
/** COL 2 : "T" LINE CODE
/** COL 3 : CODE FOR REPORT TO BE PRINTED
/** "V" : VALIDATION OF SSADM ENTITIES
/** "1" : CROSS-BOUNDARIES DATA FLOWS
/** WITHIN A DFD
/** "2" : OPERATIONAL MASTERS WITHIN A DSD
/** "3" : ALL ENTITIES WITH THEIR ATTRIBUTES
/** COL 4 : (6 CAR.) ENTITY CODE
/** (REQUIRED FOR "1" OR "2")
/**
/** -----
/**
/**BVPSADM PROC BASE=$BASE, CODE OF DEVPT DATABASE
/** INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
/** INDSN='$INDSN', INDEX OF SYSTEM NON-VSAM FILES
/** INDUV='$INDUV', INDEX OF USER VSAM FILES
/***: VSAMCAT='$VCAT', USER VSAM CATALOG
/***: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
/** STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/** USER=USER1, USER CODE (DSN SUFFIX)
/** LSR='BLSR', LSR BATCH SYSTEM NAME
/** OUT=$OUT, OUTPUT CLASS
/** UWK=$UWK WORK UNIT
/**INPUT EXEC PGM=BVPTU001
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/** DD DSN=$BCOB,DISP=SHR
/**CARTE DD DDNAME=SYSIN
/**PAC7MB DD DSN=&&SADMMB,DISP=(,PASS),
/** UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
/** DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
/**-----
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/** DD DSN=$BCOB,DISP=SHR
/**SYSOUT DD SYSOUT=&OUT
/**PACRIN DD DSN=&INDSN..BVPSY(DFSYSYPAF),DISP=SHR
/**PACROU DD DSN=&&DFSYSYPAF,DISP=(,PASS),SPACE=(TRK,1),
/** UNIT=&UWK,

```



```

//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYPAF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//PADM10 EXEC PGM=BVPADM10
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/*:          DD DSN=&SYSTCAT,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7MB DD DSN=&&SADMMB,DISP=(OLD,PASS)
//PAC7EJ DD SYSOUT=&OUT
//PAC7ET DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
/*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)

```

YSMC - YSM Methodology / WorkStation

YSMC - Introduction

This procedure is available to the users who have purchased the YSM Methodology Pacdesign module.

- It checks the validity and the integrity of the entities uploaded from the WorkStation to the Host Specifications Dictionary by the user.
- It checks the consistency between a Data flow Diagram and its parent diagram.
- It establishes different hierarchical lists of certain entities of the Database.

Note

The YSM Methodology and the procedure's functions are available in the English version only.

For complete details, refer to the 'Pacdesign' manual.

Execution conditions

None.

YSMC - User Input

One '*'-line for library access (required):

| Position | Length | Value | Meaning |
|----------|--------|----------|--|
| 2 | 1 | '*' | Line code |
| 3 | 8 | uuuuuuuu | User code |
| 11 | 8 | pppppppp | User password |
| 19 | 3 | bbb | Code of the selected library |
| 22 | 4 | nnnn | Session number (space = current) |
| 26 | 1 | 'T' | Session status if Test session |
| 37 | 25 | | Only for IMS: Request identifier (cf. PAF batch IMS) |

Entity validation request line (optional):

| Position | Length | Value | Meaning |
|----------|--------|-------|------------------------------|
| 2 | 1 | 'T' | Line code |
| 3 | 1 | | Report code: |
| | | 'W' | 'Validation of YSM entities' |

PRC entity control request lines (optional):

| Position | Length | Value | Meaning |
|----------|--------|--------|--------------------------------------|
| 2 | 1 | 'T' | Line code |
| 3 | 1 | | Report code: |
| | | 'Y' | 'Inter process consistency checking' |
| 4 | 6 | eeeeee | Entity code (PRC) |

Print-request lines (optional):

| Position | Length | Value | Meaning |
|----------|--------|--------|---|
| 2 | 1 | 'T' | Line code |
| 3 | 1 | | Report code: |
| | | '0' | 'List of Relationships' |
| | | '4' | 'Process Decomposition list (CTX)' |
| | | '5' | 'Process Decomposition list (DFD)' |
| | | '6' | 'Datastore Decomposition list' |
| | | '7' | 'Event flow Decomposition list' |
| | | '8' | 'Group Data flow Decomposition list' |
| | | '9' | 'Multiple Data flow Decomposition list' |
| 4 | 6 | eeeeee | Entity code (REL/CTX/PRC/DST/EFL/DFL) |

Printed report

This procedure prints:

- A 'Validation of YSM entities' report.
- An 'Inter-process consistency check' report.
- The reports:
 - 'List of relationships'.
 - 'Process decomposition list (CTX)'.
 - 'Process decomposition list (DFD)'.
 - 'Data store decomposition list'.
 - 'Event flow decomposition list'.
 - 'Group Data flow Decomposition list'.
 - 'Multiple Data flow Decomposition list'.

YSMC - Description of Steps

Input recognition: PTU001

Preparation of the KSDS work-file DEFINE: PRMSYS

This program is used to parameterize the KSDS work file name. The file DELETE/DEFINE is found in the DFSYSPAF member of the SY parameters PDS. Its Dsname is suffixed with the &USER parameter. The PRMSYS program sets this parameter to the value entered at execution time. The purpose is to allocate different work files from one execution to another and to avoid any conflicts when allocating work files in case of simultaneous executions.

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--------------------------------|
| PACRIN | &INDSN..BVPSY(DFSYSPAF) | Input | File DEFINE skeleton |
| PACROU | &&DFSYSPAF | Output | File DEFINE (SYSIN for IDCAMS) |

Definition of KSDS work file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DFSYSPAF | Input | File DEFINE |

Validation of YSM entities: PYSMCC

| Code | Physical name | Type | Label |
|--------|----------------------|--------------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7MB | &&YSMCMB | Input | User input |
| SYSPAF | &INDUV..SYSPAF.&USER | Input/Output | PAF standard KSDS file |
| PAC7EJ | | Report | Integrity checking lists |

| Code | Physical name | Type | Label |
|--------|---------------|--------|--------------------|
| PAC7EI | | Report | Validation reports |
| PAC7DD | | Report | Error list |

Validation of entities: PYSMC3

| Code | Physical name | Type | Label |
|--------|----------------------|--------------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7MB | &&YSMCMB | Input | User input |
| SYSPAF | &INDUV..SYSPAF.&USER | Input/Output | PAF standard KSDS file |
| PAC7EJ | | Report | Integrity checking Lists |

Validation of entities (2): PYSMC2

| Code | Physical name | Type | Label |
|--------|----------------------|--------------|------------------------------------|
| PAC7AE | &INDSV..BVP AE | Input | Error messages |
| PACGGN | &INDSV..BVP GN | Input | Administration Database Index file |
| PACGGR | &INDSV..BVP GR | Input | Administration Database Data file |
| PACGGU | &INDSV..BVP GU | Input | Administration Database Users |
| PAC7AR | &INDUV..&BASE.AR | Input | Development Database Data file |
| PAC7AN | &INDUV..&BASE.AN | Input | Development Database Index file |
| PAC7MB | &&YSMCMB | Input | User input |
| SYSPAF | &INDUV..SYSPAF.&USER | Input/Output | PAF standard KSDS file |

| Code | Physical name | Type | Label |
|--------|---------------|--------|--------------------------|
| PAC7EJ | | Report | Integrity checking lists |

Preparation of the KSDS work-file DELETE: PRMSYS

This program's purpose is to parameterize the work file name. The DELETE of the file is found in the DFSYSPAF member of the SY parameters PDS.

| Code | Physical name | Type | Label |
|--------|-------------------------|--------|--------------------------------|
| PACRIN | &INDSN..BVPSY(DLSYSPAF) | Input | File DELETE skeleton |
| PACROU | DSN=&&DLSYSPAF | Output | File DELETE (SYSIN for IDCAMS) |

Deletion of the KSDS work-file: IDCAMS

| Code | Physical name | Type | Label |
|-------|---------------|-------|-------------|
| SYSIN | &&DLSYSPAF | Input | File DELETE |

YSMC - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - PACDESIGN YSM INTEGRITY CHECKING -
/**
/** -----
/**
/** THIS PROCEDURE IS SUPPLIED FOR USERS OF THE WORKSTATION
/** AND THE YSM PACDESIGN APPLICATION METHODOLOGY.
/**
/**
/**      IT CHECKS THE VALIDITY AND THE INTEGRITY OF THE
/** ENTITIES UPLOADED FROM THE WORKSTATION TO THE HOST
/** SPECIFICATIONS DICTIONARY BY THE USER.
/**      IT CHECKS THE CONSISTENCY BETWEEN A DATA FLOW
/** DIAGRAM AND ITS PARENT DIAGRAM.(PRC)
/**      IT ESTABLISHES DIFFERENT HIERARCHICAL LISTS OF
/** CERTAIN ENTITIES OF THE DATABASE.
/** -----
/**
/**/BVPYSMC PROC BASE=$BASE,                CODE OF DEVPT DATABASE
/**      INDSV='$INDSV',                      INDEX OF SYSTEM VSAM FILES
/**      INDSN='$INDSN',                      INDEX OF SYSTEM NON-VSAM FILES
/**      INDUV='$INDUV',                      INDEX OF USER VSAM FILES
/***:    VSAMCAT='$VCAT',                     USER VSAM CATALOG
/***:    SYSTCAT='$SCAT',                     SYSTEM VSAM CATALOG
/**      STEPLIB='$HLQ..SBVPMBR8',           LIBRARY OF LOAD-MODULES

```

```

//      USER=USER1,                USER CODE (DSN SUFFIX)
//      LSR='BLSR',                LSR BATCH SYSTEM NAME
//      OUT=$OUT,                  OUTPUT CLASS
//      UWK=$UWK                    WORK UNIT
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&YSMCMB,DISP=(,PASS),
//      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYPAF,DISP=(,PASS),SPACE=(TRK,1),
//      UNIT=&UWK,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYPAF,DISP=(OLD,DELETE)
//MAXKEY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//PYSMCC EXEC PGM=BVPYSMCC
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//GNLSR DD DSN=&INDSV..BVP GN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVP GR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVP GU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//PAC7MB DD DSN=&&YSMCMB,DISP=(OLD,PASS)
//PAC7EI DD SYSOUT=&OUT
//PAC7EJ DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT

```

```

//SYSUDUMP DD SYSOUT=&OUT
//PYSMC3 EXEC PGM=BVPYSMC3,COND=((0,NE,INPUT),(0,NE,PYSMCC))
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//PAC7MB DD DSN=&&YSMCMB,DISP=(OLD,PASS)
//PAC7EJ DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PYSMC2 EXEC PGM=BVPYSMC2,COND=((0,NE,INPUT),(0,NE,PYSMCC))
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//GNLSR DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//SYSPAF DD DSN=&INDUV..SYSPAF.&USER,DISP=SHR
//PAC7MB DD DSN=&&YSMCMB,DISP=(OLD,PASS)
//PAC7EJ DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)

```




Part Number: DEPCI003359A - 8794

Printed in USA