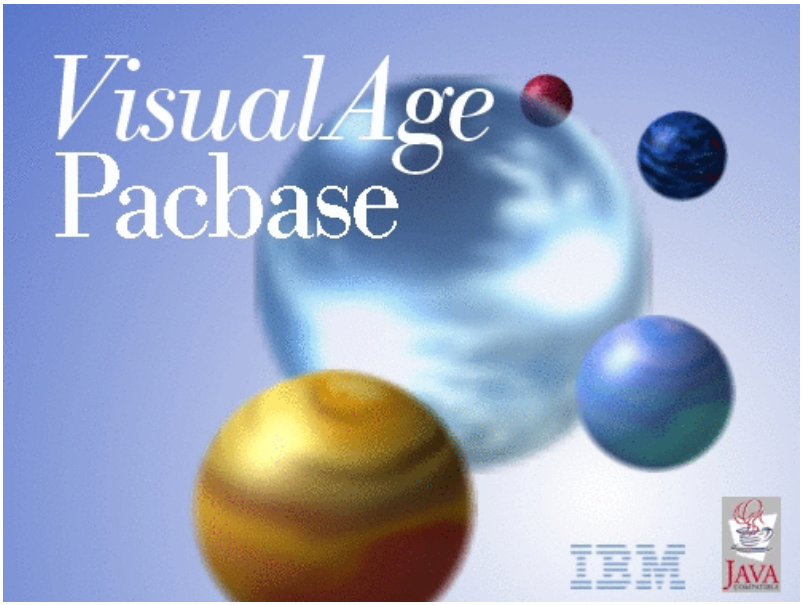
VisualAge Pacbase

DMSII DATABASE DESCRIPTION

Version 3.5



VisualAge Pacbase

DMSII DATABASE DESCRIPTION

Version 3.5

Note

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

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

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

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

First Edition (May 2007)

This edition applies to the following licensed programs:
- VisualAge Pacbase Version 3.5

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

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

Contents

Notices

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

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

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

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

Trademarks

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

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

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

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

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

Chapter 1. Introduction

VisualAge Pacbase Functions

VisualAge Pacbase Products

VisualAge Pacbase is a modular AD solution which is composed of two main products - Pacdesign for application design, Pacbench for application development.

Pacdesign and Pacbench are used to populate the Specifications Database and to ensure the maintenance of existing applications. Each product includes several functions.

Basic Functions

Dictionary

Structured Code

Personalized Documentation Manager (PDM-PDM+)

Generators

On-Line Systems Development

Pacbench Client/Server

Batch Systems Development

COB / Generator

Database Description

DBD

DBD-SQL

Application Revamping

Dialog Web Revamping

Quality Control

Pacbench Quality Control (PQC)

Quality Control Extensibility

Table Management

Pactables

Production Turnover and Follow-up

Support of Configurations Management (SCM)

Pac/Transfer

Development Support Management System (DSMS)

Additionnal services

Pac/Impact

Dictionary Extensibility

Pacbase Access Facility (PAF-PAF+)

DSMS Access Facility (DAF)

Methodology (Merise, YSM, etc.)

Sub-networks comparison utilities

Rename/move entity utility (RMEN)

Journal Statistics utility (ACTI)

RACF / TOPSECRET Security Interface

ENDEVOR

Presentation of D.B.D. Function

The Database Description function automatically generates Database descriptions adapted to the Database Management System in use. This is done by using Segment and relationship Descriptions defined during the application analysis phase.

The DBD function can generate the description of the following DBMS's:

- Relational databases,
- Network databases (CODASYL),
- Hierarchical databases (DL/1),
- Physical File - AS/400 databases and TANDEM DDL,
- DMSII databases.

Each one of these DBMS's is documented in a specific Manual.

SQL RELATIONAL DBD

This Function can only be used in conjunction with the Dictionary: data defined in the Specifications Dictionary can be used to generate Database Descriptions.

This information is described through a Database Description Language which is independent of the DBMS in use. This enables you to generate different descriptions from the same source.

Principles of Description

In this manual, the entities and screens managed by VisualAge Pacbase are described in two parts:

- An introduction which explains the purpose and the general characteristics of the entity or screen,
- A detailed description of each screen, including the input fields of on-line screens.

For the description of batch input, refer to the 'Developer's Procedures' manual.

All the on-line fields described in this manual are assigned an order number in the screen map. These numbers are also used in the screen description that follows.

If you use Developer workbench, refer to the on-line Help.

If you use the VisualAge Pacbase WorkStation, refer to the 'WorkStation User Interface' guide which documents the corresponding windows.

Note: Each type of Database Block has a specific description. As a result, fields may have different meanings or may not be used, depending on the type of Database Block.

Chapter 2. Use of the Function with DMSII

Introduction

This reference manual describing the DMSII database is not meant to be a training manual of DMSII techniques.

Initial knowledge of both DMSII and the System Specifications Dictionary function is necessary.

This manual contains many examples in order to guide the user during the realisation of a DMSII database in the system Specifications Dictionary.

The Specifications Dictionary function

The objective of the Specifications Dictionary is to manage logical descriptions of the various external views. In order to achieve this objective, it uses the following Sysyem entities:

- Data Elements,
- Segments,
- Database blocks,
- Generation Elements (-GG) associated to Database blocks.

EQUIVALENT TERMINOLOGY

Note:

A DMSII structure is a set of components, each one with its own syntax. A ″dataset″ is made of ″items″.

A Database Block calls Segments, each Segment is a list of Data Elements.

The equivalent terminology is illustrated in the following chart:

DMSII DATA	PACBASE EQUIVALENT
Structure	Database Block
Dataset	Segment
Data Item	Data Element
Component : Dataset, Access, Set	Description Line
Subset, Link, Remap	

First of all, to build a DMSII structure, the called DMSII entities must be defined and described. The description performs the calls of these entities components.

The Data Elements generating DMSII data must initially be defined in the System Specifications Dictionary.

Principle of Utilization

A Database Block allows the generation of a DMSII structure. The System D.A.S.D.L. extracts all the information initially entered in the Specifications Dictionary (logical level information).

This information comes from the definition lines, the description lines and the Generation Elements lines associated with the Database Blocks.

From a description line, the System can find a Segment description and the Data Elements which belong to to it.

Chapter 3. ″DATA ITEM″ = DATA ELEMENT

Definition of an ″ITEM″ (E.....)

An Item is comparable to a Data Element and can be defined by a Data Element definition line. This line can be accessed with the choice:

CH: E.....

FORMATS

The System generates the data type and length from the internal format.

The types which can be directly interpreted are:

-Alpha : non numerical Display usage (D),

-Numerical : NUMBER usage (N),

-Real : BINARY usage (P).

NOTE: The System usage is indicated between brackets.

All the System formats lead to a generation. An error message appears at the end of the DASDL and points out the non-standard cases. It is the user's responsability to check the compatibility of the result.

The boolean type and boolean field do not correspond to the System formats. However, it is possible to get these types in the Segment description. Then, the DMSII Database can include boolean data.

```
                                                                    *PDMCA.PDEV.HP3.851

DATA ELEMENT CODE   1  AGE

NAME...............:2  AGE
TYPE...............:3  R

INPUT FORMAT.......:5  9(2)                                  LENGTH...:     2
INTERNAL FORMAT....:6  9(2)                  USAGE :7  N     LENGTH...:     2
OUTPUT FORMAT......:8  9(2)                       Z:9        LENGTH...:     2

EXPLICIT KEYWORDS..: 10

PARENT ELEMENT.....: 11




UPDATED BY.........:             ON:              AT:   :  :      LIB:
SESSION NUMBER.....: 0851        LIBRARY......: HP3    LOCK....:


O: C1 CH: EAGE                         ACTION:
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		DATA ELEMENT CODE (REQUIRED)
			Enter the mnemonic code which references the Data Element independently of any Data Structure, Report or Screen to which the Data Element might belong.
			There is no need to include a Report, Screen or Segment code in the Data Element code since the System does it automatically.
			This code consists of alphabetic or numeric characters only.
			Some Data Element codes are reserved by the System for use in Data Structures, Reports or Screens and cannot be defined in the Specifications Dictionary:
		'SUITE'	Prohibited. This code is reserved for the System for program generation.
		'FILLER'	Data Element that is used for the alignment of fields.
			Options of the BSD Function:
			Error Verification fields on transaction files:
		'ENPR' 'GRPR' 'ERUT'	Used for Data Element error verification. Used for Segment error verification. Used for user defined errors.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			For more information see DATA ELEMENT CODE on the Segment Call of Elements.
			For Reports:
		'LIGNE'	Reserved for the placement and alignment of the layout line. It is used only for a '00' structure.
		'LSKP'	Reserved usage only in the '00' Report Structure. See STRUCTURE NUMBER on the Report Call of Elements.
		'SAUT'	Reserved usage. This code is the counterpart of LSKP and used with the French version of the System.
			Options of the OLSD and Pacbench C/S (TUI Client) Functions:
		'ERMSG'	Data Element for the placement of the error message.
		'LIERR'	Reserved usage. This code is the counterpart of ERMSG and used with the French version of the System.
		'PFKEY'	Used to represent the programmable function keys.
		'*PASWD'	(IMS only): Used for passwords on a specific screen.
			For more information see DATA ELEMENT CODE OR SCREEN CODE TO CALL on the Call of Elements.
2	36		NAME OF DATA ELEMENT (REQUIRED IN CREAT)
			This name should be as explicit as possible. Words used here become implicit keywords (subject to limitations specified in the Character-Mode User Interface Guide, chapter 'Search for Instances', subchapter 'Searching by Keywords').
			This name appears in documentation and in Volumes each time the Data Element is used. It is also possible to list Data Elements sorted by name.
			In IMS: Use uppercase.
3	1		TYPE
		'P'	Property: Elementary piece of information defined at the conceptual level. Note: the FORMAT is optional.
		'R'	Real Data Element (Default value): elementary piece of information, defined at the Specifications Dictionary level.
			D.B.D. function: CODASYL elementary data, Relational column.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'A'	ALIAS Data Element: This value is used in conjunction with the 'A*' value in the DATA STRUCTURE CODE IN GENER. DESCR. field with the 'DATA' PIA, causes the NAME OF DATA ELEMENT to be generated, rather than the standard element name.
		'L'	Large Object Data Element
4	10		INPUT FORMAT
			Not used with the DBD function.
5	10		Internal format
			Format normally used in system files (permanent, database and temporary files) and in screen input fields.
			Like the INPUT FORMAT, the INTERNAL FORMAT will be automatically used in the data Segment descriptions.
			For batch Programs, the user may select the format type on the Program Call of Data Structures (-CD) screen.
			It is also used (with the necessary transformations) in screen descriptions (input fields). (Refer to screen description in the 'On-Line Systems Development' Manual and 'Pacbench C/S: Business Logic and TUI Clients' (chapter 'TUI Clients')).
			The internal format must be coded like a COBOL picture (without print characters).
			The 'INTERNAL USAGE' clause is associated with this format.
			Data Elements that represent a date can be assigned a symbolic format:
			Display type formats (input):
		D	Without century (DDMMYY or MMDDYY).
		C	With century (DDMMCCYY or MMDDCCYY).
			Internal type formats:
		I	Without century (YYMMDD).
		S	With century (CCYYMMDD).
			Extended type formats (output) (with slashes):
		E	Without century (DD/MM/YY or MM/DD/YY).
		M	With century (DD/MM/CCYY or MM/DD/CCYY).
		G	Gregorian format (CCYY-MM-DD).
		T	TIME format (HH:MM:SS).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		TS	TIMESTAMP format
			METHODOLOGY function: This field may be left blank for a property.
			For the formats which include a separator (E, G, M and T), you can specify, after the character which represents the format, a separator if you do not want to use the separator included by default in the format (For example, A 'G/' format will generate CCYY/MM/DD instead of CCYY-MM-DD, which is the default Gregorian format).
			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" Manual.
6	1		INTERNAL USAGE
			Corresponds to the COBOL 'USAGE' clause.
		'D'	DISPLAY (default option), all hardware. Required for data elements indicating dates.
		'C'	COMPUTATIONAL (binary), IBM or equivalent; COMPUTATIONAL-4 (binary), IBM SYSTEM 38; COMPUTATIONAL-4 IBM 3-15D, COMPUTATIONAL-6 ICL 2900; BINARY, IBM and COBOL II variant.
		'R'	COMPUTATIONAL SYNCHRONIZED RIGHT, IBM or equivalent; This value is preferable to 'C' when binary data is aligned on even addresses, since corresponding COBOL statements are more efficient.
		'B'	COMPUTATIONAL-1 ICL 1900. BINARY-1 UNISYS 1100 associated with format 1(n).
		'S'	COMPUTATIONAL SYNCHRONIZED RIGHT ICL 1900.
		'N'	COMPUTATIONAL-4 aligned on a half-byte. You must add the complement if the length is uneven.
		'P'	COMPUTATIONAL-1 BULL 66, 6000 and DPS8.
		'L'	COMPUTATIONAL-1 SYNCHRONIZED RIGHT ICL 1900.
		'Q'	COMPUTATIONAL BULL 66, 6000 and DPS8.
		'F'	COMPUTATIONAL-1 IBM or equivalent. COMPUTATIONAL-9 BULL DPS7. COMPUTATIONAL-11 BULL 66 and DPS8. Relational DBD : floating point, simple precision.
		'T'	COMPUTATIONAL-3 PACKED SYNC. BULL 66 and DPS8.
		'X'	DISPLAY SIGN IS TRAILING SEPARATE CHARACTER.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'G'	COMPUTATIONAL SYNCHRONIZED RIGHT ICL 2900 AND COMPUTATIONAL-5 MICROFOCUS.
		'7'	COMPUTATIONAL-5 ICL 2900.
		'K'	COMPUTATIONAL CDC. COMPUTATIONAL UNISYS 1100 (COBOL 85)
		'M'	COMPUTATIONAL-1 CDC.
		'N'	COMPUTATIONAL UNISYS-A
		'O'	COMPUTATIONAL-4 UNISYS 1100
		'U'	COMPUTATIONAL-1 UNISYS 1100.
		'W'	COMPUTATIONAL-2 UNISYS 1100. COMPUTATIONAL-12 BULL 66 and DPS8. RELATIONAL DBD : floating point, double precision.
		'H'	COMPUTATIONAL UNISYS 1100. BINARY UNISYS 1100 (COBOL 85)
		'8'	COMPUTATIONAL BULL 66 COBOL 74 and DPS8.
		'9'	COMPUTATIONAL-3 BULL 66 COBOL 74 DPS7 and DPS8.
		'J'	COMPUTATIONAL-6 BULL 66 COBOL 74 DPS7 and DPS8. REAL UNISYS-A.
		'Y'	DB-KEY BULL 66 DM4 and DPS8. POINTER IBM and MICROFOCUS.
		'I'	DISPLAY-1 Unisys 1100
		'5'	COMPUTATIONAL-1 BULL 64 66 MINI-6 COBOL 74 DPS7 DPS8
		'6'	COMPUTATIONAL-2 BULL 64 66 MINI-6 COBOL 74 DPS7 DPS8
		'3'	COMPUTATIONAL-3 IBM or equivalent. COMPUTATIONAL BULL 64 MINI-6 DPS7. COMPUTATIONAL-3 (packed decimal) IBM SYSTEM 38. PACKED-DECIMAL UNISYS 1100 (COBOL 85)
		'0'	COMPUTATIONAL-7 BULL 66 and DPS8.
		'1'	DISPLAY-1 NCR (signed extended decimal). DISPLAY SIGN LEADING SEPARATE - UNISYS 1100, DPS8, IBM, TANDEM, DPS7.
		'4'	DISPLAY-2 NCR (unsigned packed decimal).
		'2'	DISPLAY-2 BULL = DISPLAY, fields are compared in accordance with the "commercial collating sequence" and not in accordance with the standard BULL sequence.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'Z'	In batch mode only: this option, which is only used with an output format, allows for the generation of a 'BLANK WHEN ZERO' clause with the Batch D.S. function.
			METHODOLOGY function: This field may be left blank for a property.
7	27		OUTPUT FORMAT
			Not used by the DBD function.
8	1		BLANK WHEN ZERO CLAUSE
			This field is not used when defining a data element used to generate a CODASYL elementary data element or a relational column.
9	55		EXPLICIT KEYWORDS
			This field allows you to enter additional (explicit) keywords. By default, keywords are generated from the instance's name (implicit keywords).
			Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage and are therefore ignored in keywords.
			Keywords are not case-sensitive: uppercase and lower-case letters are equivalent.
			NOTE: Accented and special characters can be declared as equivalent to an internal value in order to optimize the search of instances by keywords (Administrator workbench, 'Window' menu, 'Parameters browser' choice, in 'Special Characters' tab).
			A maximum of ten explicit keywords can be assigned to one entity. For more details, refer to the 'Character Mode User Interface' guide, chapter 'Search for Instances', subchapter 'Searching by Keywords'.
10	6		PARENT ELEMENT CODE
			Allows Data Elements sharing the same characteristics to be defined under different codes.
			If a parent Data Element is indicated, the Data Element takes on the characteristics of the parent by default. These can be modified at the child level.
			The parent Data Element must have been defined previously.

Chapter 4. ″DATASET″ = SEGMENT

Definition of a ″DATASET″ (S....)

A ″Dataset″ is similar to a segment and is defined by a Segment definition line. This line can be accessed with the following choice:

CH: S....

A Segment can be used to generate either a ″dataset″ or a ″remap″. The type of generation a Segment will perform is defined when it is called in the Database.

PREREQUISITES

The Data Structure on which the Segment depends must be defined.

```
                                                                    *PDMCA.PDEV.HP3.851
                        1 2
SEGMENT CODE           DL40


NAME.....................:3 PERSONNELS
OCCUR. OF SEGMENT IN TABLE:4
EST. NUMBER OF INSTANCES..:5



VALUE OF RECORD TYPE ELEM.:6
CODE OF ACTION CODE ELEM..:7
PRESENCE..................: CR:       MO:       DE:
                           M4:       M5:       M6:


EXPLICIT KEYWORDS..:8


UPDATED BY.........:             ON:             AT:   :  :      LIB:
SESSION NUMBER.....: 0851        LIBRARY......: HP3    LOCK....:


O: C1 CH:  SDL40                       ACTION:
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			DATA STRUCTURE / SEGMENT CODE
1	2		DATA STRUCTURE CODE (REQUIRED)
			This code is made up of two alphanumeric characters. This is a logical code internal to the Database and therefore independent of the names used in Database Blocks and Programs.
2	2		Segment number (REQUIRED)
			The first character must be numeric and the second either numeric or alphabetic. However the second character can be alphabetic only if the first character is other than zero.
		00	For standard files:
			Used to indicate the common part of records in a file, located at the beginning of each record (Default).
			The control break sort keys, the record type and the keys of indexed files are contained in this Segment.
			A file does not necessarily have a common part.
			Records on files with only one type of record should be coded as a '00' Segment.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			With the Pactables function, this value is not allowed.
		01-99	Designates a specific Segment. The common part Data Elements are automatically concatenated with each specific part Segment. Although a data element may not be used twice in the same Segment, it may be used in both the common part and in one or more specific Segments (except data structures used as Tables).
3	36		SEGMENT NAME (REQUIRED IN CREAT)
			This name must be as explicit as possible because it is used in the automatic building of keywords, Words used here become implicit keywords (subject to limitations specified in the Character-Mode User Interface Guide, chapter 'Search for Instances', subchapter 'Searching by Keywords').
4	4	NUMER.	Occurrences of segment in table
			PURE NUMERIC FIELD
			BATCH SYSTEMS DEVELOPMENT:
			This is the amount of space reserved for a Segment in memory (USAGE OF DATA STRUCTURE 'T' or 'X', or RECORD TYPE = 3, or 4.
			For tables (USAGE OF DATA STRUCTURE 'T' or 'X'), the default value at generation time is 100.
			Pactables:
			This field is strictly for documentation purposes.
			PACBENCH C/S:
			The value entered in this field indicates the repetitive read or update capacity of the server which calls the Logical View. This capacity is expressed by a maximum number of repetitions. The Logical View can then be used as a repeated structure.
			NOTE: The use of a Logical View in a card layout does not exclude its use in a row layout. It is therefore strongly recommended to systematically fill in this field. Moreover, the entered value must be high enough to limit the exchanges between the client and the server.
5	9	NUMER.	Estimated number of instances
			PURE NUMERIC FIELD
			For the Batch Systems Development function, this field is used to specify the estimated number of occurrences for a segment in a database or in a standard file.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			For the METHODOLOGY function, this field is used for activity calculation on the record or set using the Segment (on-line only).
			For the DBD function, this field is used to specify the application number of an entity in a SOCRATE/CLIO Block.
6	10		CODE / VALUE OF RECORD TYPE ELEMENT
			For a Relational Table or View, this field is used to specify the external name between quotes.
			This field is not used to define a CODASYL record.
7	36		CODE OF ACTION CODE ELEMENT
			This field is not used to define a CODASYL record or a Relational Table or View.
8	55		EXPLICIT KEYWORDS
			This field allows you to enter additional (explicit) keywords. By default, keywords are generated from the instance's name (implicit keywords).
			Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage and are therefore ignored in keywords.
			Keywords are not case-sensitive: uppercase and lower-case letters are equivalent.
			NOTE: Accented and special characters can be declared as equivalent to an internal value in order to optimize the search of instances by keywords (Administrator workbench, 'Window' menu, 'Parameters browser' choice, in 'Special Characters' tab).
			A maximum of ten explicit keywords can be assigned to one entity. For more details, refer to the 'Character Mode User Interface' guide, chapter 'Search for Instances', subchapter 'Searching by Keywords'.

Description of a ″DATASET″ (S....CE)

A ″Dataset″ is similar to a Segment and is described in the same way as a Segment. The description screen of a Segment is called with the following choice:

CH: S....CE

This description is the list of ″items″ in the ″Dataset″ or in the ″remap″. It is a list of calls of Data Elements in the Segments.

PREREQUISITES

The ″dataset″ and the called ″items″ must be defined.

INFORMATION RECOGNIZED

The only data to have an impact on the block generated program are the ones indicated and entered in the -CE.

-Number of repetitions: for OCCURS clause.

-Number of Data Element within a group: for GROUP type.

-Access key or sort indicator: for BOOLEAN types, FIELD, the ″remap regrouping″ and VIRTUAL. The item RECORD TYPE is also identified here.

-Presence indicator: first position for the REQUIRED clause.

-Update/table: for DEPENDING ON of OCCURS.

Notes:

.A boolean field is indicated with a GROUP and the indicator FIELD.

.In a ″remap″, a virtual boolean or field is not automatically obtained (the same field is used).

ROLE IN THE GENERATION

A Segment description is used to describe a dataset or a ″remap″. A ″remap″ is therefore a Segment in which the Data Elements chosen are called automatically. It is then impossible to use the functionality that automatically hides Data Elements (HIDDEN). A group Data Element in the dataset can be called without this notion.

Note: For the groups in a ″remap″, the number of repetitions (OCCURS) and the presence indicator (REQUIRED) are ignored.

```
                          1 2                              *PDMCA.PDEV.HP3.851
SEGMENT CALL OF ELEMENTS DL40 PERSONNELS
3 4     5      7          8 9   10 11  12   <      13    >  14        15
A LIN : ELEM.  INT.FORM. U OCC GR K CMD456 CONT VALUE/SFC  UPD/TRGET DOC LIB
  100 : NBPER                                                        0851
  200 : NOMC                    2                                    0851
  210 : NOM                                                          0851
  220 : PRENOM                                                       0851
  250 : SEXE                      B                                  0851
  260 : AGE                                                          0851
  270 : SSNO                        0                                0851
  300 : DPT                                                          0851
  310 : RANG                                                         0851
  320 : SALAIR                                                       0851
  400 : IDCOUR               8                                       0851
  500 : TELEPH                                                       0851
  600 : SUPER                                                        0851
      :
      :
      :
      :
      : NAME    : 6
*** END ***
O: C1 CH:  -CE
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			DATA STRUCTURE / SEGMENT CODE
1	2		DATA STRUCTURE CODE (REQUIRED)
			This code is made up of two alphanumeric characters. This is a logical code internal to the Database and therefore independent of the names used in Database Blocks and Programs.
2	2		Segment number (REQUIRED)
			The first character must be numeric and the second either numeric or alphabetic. However the second character can be alphabetic only if the first character is other than zero.
		00	For standard files:
			Used to indicate the common part of records in a file, located at the beginning of each record (Default).
			The control break sort keys, the record type and the keys of indexed files are contained in this Segment.
			A file does not necessarily have a common part.
			Records on files with only one type of record should be coded as a '00' Segment.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			With the Pactables function, this value is not allowed.
		01-99	Designates a specific Segment. The common part Data Elements are automatically concatenated with each specific part Segment. Although a data element may not be used twice in the same Segment, it may be used in both the common part and in one or more specific Segments (except data structures used as Tables).
3	1		ACTION CODE (REQUIRED)
		'C'	Creation of the line
		M	Modification of the line
		D or 'A'	Deletion of the line
		T	Transfer of the line
		B	Beginning of multiple deletion
		G	Multiple transfer
		?	Request for HELP documentation
		E or '-'	Inhibit implicit update
		X	Implicit update without upper/lowercase transformation.
4	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.
5	6		DATA ELEMENT CODE
			ELEMENTARY DATA ELEMENT DEFINED IN THE DICTIONARY
			The Data Element automatically assumes the characteristics defined at the Specifications Dictionary level.
			If the Data Element is used as a group, its format depends on the characteristics of the elementary Elements that make up the group.
			If the group is used as a key (sort or access key), the composite format of the elementary Elements must be compatible with the format specified for the group.
			DATA ELEMENT NOT DEFINED IN THE DICTIONARY

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			The name and/or format of undefined Data Elements must be indicated at the segment level.
			RESERVED DATA ELEMENT CODES
		SUITE	Prohibited. This code is reserved for the System for program generation.
		FILLER	Data Element that is used for the alignment of fields.
			OPTIONS OF THE BATCH SYSTEMS DEVELOPMENT FUNCTION
			These codes (when used) precede other entries made in this field, in the sequence described below.
		ENPR	Used to store Element error verifications in a transaction file. The length is n + 1 where n = either the total number of elementary Elements in the file, or the number of elementary Elements in the '00' Segment added to the largest non-00 Segment. ("Largest" here means the most elementary Elements.) This depends upon the value entered in the RESERVED ERROR CODES IN TRANS FILE field on the Call of Data Structures (-CD) screen.
		GRPR	Used to store Segment error verifications. Its length is n + 1 where n = the number of records.
		ERUT	Used to store error verifications for users.
			Normally, these last three Data Elements are used in transaction files for error verification fields. When used in other types of files as "optional" Data Elements, they may be used as group fields whose generation may be invoked or suppressed according to the option selected in the RESERVED ERROR CODES IN TRANS. FILE field. (Note: this will affect the elementary Elements within the group as well.)
			CALLING DATA AGGREGATES
			A SEGMENT CODE or a Model Entity code (Relationship or Object in the METHODOLOGY function) can be entered in this field. The called data aggregate will be interpreted as if the individual Elements that make it up had been entered.
			The NO. OF ELEMENTARY ELEMENTS IN GROUP field is used to identify data aggregate calls.
			Enter the code at the location the elements are to be included in the Segment description.
			In O:C2, the level of 'nesting' is displayed in the Action Code (up to four levels).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			The number of authorized nesting levels varies according to the type of generator. Up to 4 nesting levels are authorized for data generation and PAF use.
			CONTINUATION LINES
			It is possible to create continuation lines. This may be necessary if there are many validations on a Data Element. In this case, leave the DATA ELEMENT CODE field blank, and use a LINE NUMBER value that sequentially follows that of the line where the Data Element code was entered.
			The continuation lines are taken into account if the Data Element code is blank or if the Data Element code is the same as the previous one.
6	18		NAME OF DATA ELEMENT
			It is not required for a Data Element which is not defined in the Data Dictionary.
			However, it is optional for a data aggregate or a FILLER.
			NOTE: For on-line entry of Data Elements that are not declared in the Dictionary, this field cannot be used to input more than one Data Element at a time. There is actually only one available field on this screen, whether for input or for display.
			To define an Element at the Segment level :
			- Enter the Element code (and possibly the format) on the -CE, line nnn,
			- On the 'name' line, repeat the line number (nnn), and indicate the name (18 characters maximum),
			- Use the C2 option to view the name and format.
			NOTE: If several undefined Data Elements have been defined in the Dictionary, only the name of the first Data Element will be displayed if the Choice 'CH:S.....CE' is used.
			To view the name of the Data Element CODEL, on line 130, for example, use the choice 'O: C2 CH: Sssss-CE130'. This will display the Data Elements called in the Segment 'ssss' from the line 130 on.
7	10		Data element internal format
			It is required only in the following cases :
			- For an elementary Data Element not defined in the Dictionary (COBOL format),

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			- For a group Data Element that is or belongs to a key; its length must be the sum of the lengths of its elementary Data Elements,
			- For a FILLER-type field.
			It is the internal format; input and output formats will be the same (but with usage Display). It is defined as on a Data Element Definition screen.
8	1		INTERNAL USE
			For Data Elements not defined in the Specifications Dictionary when the INTERNAL FORMAT OF DATA ELEMENT field has been given a value, enter the appropriate USAGE (default : 'D' for DISPLAY).
			For valid values, see the USAGE field on the Data Element Definition Screen.
9	3		OCCURRENCES (COBOL "OCCURS" CLAUSE)
			PURE NUMERIC FIELD
			This field represents the 'OCCURS' clause at an elementary Data Element level, or at a group level (Maximum of 3 levels).
			It can be changed into an 'OCCURS DEPENDING ON' clause by entering '**' in the UPDATE TARGET field, followed by the counter's Segment and Data Element codes.
			The COBOL restrictions on the OCCURS clause apply.
10	2		No. of elementary elements in group
			PSEUDO NUMERIC FIELD
		'1 to 99'	For group Data Elements, enter the number of elementary Elements that belong to the group (A Segment call is considered as an elementary Data Element).
			Groups may contain up to 99 elementary Elements. Group Elements may contain embedded groups however the total number of elementary Elements cannot exceed 99. (The group Data Element codes are not counted). The maximum number of levels of 'nesting' is 9.
			This field is also used to identify the entity called in the DATA ELEMENT CODE field as Methodology entities or previously defined Segments.
		'*M' '**'	Call of an Object or a Relationship. Call of a Segment.
		'**'	SQL DBD function: Call of a Segment into a view.
11	1		ACCESS KEY OR SORT KEY

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			It allows to manage with DMSII information that PACBASE entities do not know and to get a generation that includes these particularities.
			″Item″ type not defined in the Data Element:
		’B’	.Boolean
		’F’	.Field : integer or boolean field
		’T’	.Record Type : identifier of the variable part
			For the ″remap″:
		’V’	.virtual item
		’R’	.group defined in the ″remap″ (It is the case for ″remap regrouping″)
12	6		DATA ELEMENTS PRESENCE CONTROL
			Only the field first position is used to indicate the presence of an ″item″ in a ″dataset″.
		Blank	Optional ″item″ (default value).
		’O’	″Item″ REQUIRED.
			This clause will be generated after the ″item″ definition in the ″dataset″.
13	14		CONT AND VALUE/SFC FIELDS
			These fields are not used.
14	16		INDICATION OF OCCURS DEPENDING ON
			-’UPD/TRGET’ FIELD
		’**’	This means that the ″occurs″ is ″DEPENDING ON″ for the Data Element in this line.
15	1		DOCUMENTATION INDICATOR
			This field is used in on-line mode only. It is a read-only field.
		’*’	A Comment, a Generation Element or an Error Message has been assigned to the element called on this line.
			Access to line nnn: -CEnnn, or -Dxnnn for a Database Block (with x = C, H or R depending on the Block type)
			To access the Comment, Generation Element or Error Message assigned to the called element, enter the access to line nnn followed (without blank) by GC (for Comment), GG (for Generation Element) or GE (for Error Message).

Chapter 5. DATABASE = BLOCK

Definition of a DATABASE (B.....)

A DMSII Database is similar to a System Block and is defined by a Database Block definition line.

A Database Block is defined with a code, a name and a type.

A Database Block used to generate a DMSII structure is to be defined by type ″20″.

```
                                                                    *PDMCA.PDEV.HP3.851
                                    1
BLOCK  CODE                      TDASDL

NAME..................:2 TEST GENERATION DASDL
TYPE..................:3 20 DMS II (DASDL)
VERSION...............:4

EXTERNAL NAME.........:5


CONTROL CARDS....... FRONT:6        BACK:7


EXPLICIT KEYWORDS..:8



UPDATED BY.........:             ON:             AT:   :  :      LIB:
SESSION NUMBER.....: 0851        LIBRARY......: HP3    LOCK....:



O: C1 CH:  BTDASDL                     ACTION:
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		BLOCK CODE (REQUIRED)
			One to six alphanumeric characters.
2	36		NAME OF THE BLOCK (REQUIRED IN CREAT)
			This clear name should be as explicit as possible. Words used here become implicit keywords (subject to limitations specified in Subchapter ″HOW TO BUILD THE THESAURUS″, Chapter ″KEYWORDS″ in the SPECIFICATIONS DICTIONARY Reference Manual).
3	2		TYPE OF BLOCK (REQUIRED IN CREAT)
			For hierarchical or network databases, it is not required, when creating a database block, to enter the definitive block type. The selection of a network or hierarchical structure is sufficient at this point.
			A specific ″physical″ type must be entered when generating the Data Description Language (DDL).
		'TR' 'SE'	Tree-like structure (hierarchical block). Group of sets (network block).
			HIERARCHICAL DATABASES - IMS/DL1
		'DP'	Physical Database Description.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'DR'	Physical Database Description (same as 'DP', but only the data elements referenced as access keys in the segment description are generated in the 'FIELD.....' statements).
		'DL'	Logical Database Description.
		'PC'	PCB.
		'IP'	Primary Index.
		'IS'	Secondary Index.
		'PS'	PSB (Assigned at creation. Cannot be modified at a later stage).
			RELATIONAL DATABASES
		Q2	DB2 SQL
		Q3	SQL SERVER
		QB	DB2/2 and DB2/6000
		QC	DATACOM/DB
		QN	NONSTOP SQL
		QP	ORACLE
		QR	RDMS
		QS	SQL/DS
		QT	INTEREL RDBC
		QU	INTEREL RFM
		QY	SYBASE
		DB	DB2 (It is recommended to use the Q2 type)
			NETWORK DATABASES
			.CODASYL-DM4 (GCOS8):
		'M1'	DDL schema, only elementary fields are generated,
		'M4'	DDL schema, only group fields are generated,
		'M2'	DMCL schema,
		'M3'	Sub-schema.
			.CODASYL-IDS2 (GCOS7):
		'I1'	DDL schema,
		'I2'	DMCL schema,
		'I3'	SDDL sub-schema.
			.CODASYL-IDMS:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'D0'	DDL schema (Release 10.0),
		'D1'	DDL schema,
		'D2'	DMCL schema,
		'D3'	Sub-schema,
		'D4'	Sub-schema (Release 5.7).
			.CODASYL-DMS (UNISYS 1100):
		'S1'	DDL Schema,
		'S3'	Sub-schema.
			DDL TANDEM
		TD	TANDEM
			AS/400 PHYSICAL FILE
		PF	AS/400 Physical file (IBM SYS. 38)
		LF	AS/400 Logical file (IBM SYS. 38).
			DMSII DATABASE
		20	DMSII Database (DASDL)
4	4		VERSION
			This field is not used.
5	8		DATABASE BLOCK EXTERNAL NAME
			Necessary at generation time.
			This is the physical name of the System-generated DDL (Data Description Language) module.
			To obtain a list of blocks sorted by this external name, enter 'LEB' in the CHOICE field.
			For TurboImage, only the first six characters are processed.
6	1		CONTROL CARDS IN FRONT OF BLOCK
			Necessary at generation time.
			Enter the one-character code that identifies the job control card to be inserted before the generated block.
7	1		CONTROL CARDS IN BACK OF BLOCK
			Necessary at generation time.
			Enter the one-character code that identifies the job control card to be inserted after the generated block.
8	55		EXPLICIT KEYWORDS

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			This field allows you to enter additional (explicit) keywords. By default, keywords are generated from the instance's name (implicit keywords).
			Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage and are therefore ignored in keywords.
			Keywords are not case-sensitive: uppercase and lower-case letters are equivalent.
			NOTE: Accented and special characters can be declared as equivalent to an internal value in order to optimize the search of instances by keywords (Administrator workbench, 'Window' menu, 'Parameters browser' choice, in 'Special Characters' tab).
			A maximum of ten explicit keywords can be assigned to one entity. For more details, refer to the 'Character Mode User Interface' guide, chapter 'Search for Instances', subchapter 'Searching by Keywords'.

Description of a DATABASE (B.....DC)

A DMSII Database is similar to a System Block and is described by a Database Block description line.

CH: B.....DC

This description is a list of elements within the Database.

Six line types are taken into account:

```
. Dataset       ---> 1
. Access        ---> 2
. Set           ---> 3
. Subset        ---> 4
. Link          ---> 5
. Remap         ---> 6
```

DESCRIPTION ORDER

The description lines are ordered by their number. This order corresponds to the generation presentation order. The notion of parent Segment allows interlockings management but does not interfere in the location of generated

elements. Attention must be payed to embedded elements (EMBEDDED) especially ″accesses″, sets and subsets in order to get a good generation.

LINKS

Links are not identified by a type but by the line codification.

```
In the order of the line, the following elements are found:
.Counted          : datell , ffssp , ffss , C
.Self-correction  : datell , C     , ffss , set
.Symbolic         : datell , S     , ffss , set
.Unprotected      : datell , ffssp , ffss , N
.Verified         : datell , ffssp , ffss , datelp
'Datell' is the data element code link, 'datelp' is the
data element in the reference segment, 'ffss' is the
child dataset, and 'ffssp' is the parent dataset.
Generated elements are in the ffss dataset:
.ffss-datell IS IN ffssp COUNTED;
.ffss-datell IS IN set;
.ffss-datell IS KEY OF set;
.ffss-datell IS IN ffssp WITH NO PROTECTION;
.ffss-datell IS IN ffssp VERIFIED ON ffssp-datelp;
VARIABLE STRUCTURES
```

In a Dataset there is a variable part and a fixed part.

The fixed part is indicated by a number present in the corresponding field. The item ″RECORD TYPE″ must be coded in the fixed part and is marked with the letter ″T″ in the sort key.

Each variable is identifiable by the letter ″V″ located in the ″set or Data Element code″ field. The number indicated in the corresponding column is the DMSII internal identifier of the variable record.

```
                                1                          *PDMCA.PDEV.HP3.851
DESCRIPTION OF DASDL DMSII  TDASDL TEST GENERATION DASDL
2 3     4 5      6    7    8          9   10
A LIN : T SET NA DATASET   OPTION     PT. COMMENT
      :   LIN IT EMB.      SET/IT     N K
  100 : 1             DL10 S      *       MAIN FILE : COURSES
  101 : 5 PROF   DL40 DL10 C
  200 : 1        DL10 DL20 U      *       BOOKS
  210 : 4 LIVK   DL10 DL20 LI     *
  300 : 1        DL10 DL30 S      *       STUDENTS
  301 : 5 SSNO   C    DL30 MFSSET
  302 : 5 SSNO1  S    DL30 MFSSET
  310 : 3 ETUSET DL10 DL30 IS     * 00002
  350 : 3 COUSET      DL10 IS     *
  400 : 1             DL40 S      *       PERSONNEL
  401 : 5 IDCOUR DL10 DL40 C
  402 : 5 TELEPH DL80 DL40 TELEPH
  403 : 5 SUPER  DL40 DL40 N
  410 : 3 SS-U-P      DL40 IS     *
  420 : 3 U-P-ST      DL40 IS     *
  500 : 1             DL50 S      *       REGISTERED
  501 : 5 TELEPH DL80 DL50 TELEPH

O: C1 CH:  -DC
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		BLOCK CODE (REQUIRED)
			One to six alphanumeric characters.
2	1		ACTION CODE
		'C'	Creation of the line
		M	Modification of the line
		D or 'A'	Deletion of the line
		T	Transfer of the line
		B	Beginning of multiple deletion
		G	Multiple transfer
		?	Request for HELP documentation
		E or '-'	Inhibit implicit update
		X	Implicit update without upper/lowercase transformation.
3	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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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.
4	1		LINE TYPE (REQUIRED)
			It identifies the DMSII element which is to be defined.
		1	Dataset
		2	Access
		3	Set
		4	Subset
		5	Link
		6	Remap
5	6		SET OR DATA ELEMENT CODE
			This field has three different meanings:
			.For accesses(2), sets(3) and subsets(4): This element name (access, set or subset).
			.For a link(5): The Data Element code which is a link.
		Blank, 'V'	.For dataset(1) and remap (6): The segment is comparable to a dataset or a dataset "remap". The segment describes a variable part.
6	4		PARENT SEGMENT CODE
			Indicates if the affected element is embedded.
			For a link (5):
		ffss	Indicates the segment which is the link reference for "counted, unprotected, verify link".
			Differenciates the link type:
		'C'	Self-correction link.
		'S'	Symbolic link.
7	4		SEGMENT CODE (REQUIRED)
			DMSII element or reference to this element.
8	6		DATASET, SET/DATA ELEMENT TYPE
			The meaning changes depending on the line type.
			For the dataset (1), the dataset type:
		'C'	-Compact.
		'D'	-Direct.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'O'	-Ordered.
		'R'	-Random.
		'RE'	-Restart.
		'S'	-Standard.
		'U'	-Unordered.
			For sets (3) and subsets (4), set or subset type:
		'BV'	-Vector Bit.
		'IR'	-Random Index.
		'IS'	-Sequential Index.
		'LI'	-Unordered List.
		'OL'	-Ordered List.
		Other	-Reference set for the subset.
			For links (5), three possible contents:
			-Type Distinction
		'C'	.Counted link.
		'N'	.Unprotected link.
			-Link reference set
			.Self-correction. .Symbolic link.
			-Data Element code in the reference Segment
			.Verified link.
			For a "Remap" (6):
		=ffss	The equal sign followed by the remapped Segment
			code.
9	5	NUMER.	NUMBER IDENTIFYING VARIABLE PART
			This field has different meanings depending on the line type.
			For datasets (1) and remaps (6): The variable part identifier or the "record type" maximum value.
			For accesses (2), sets (3) and subsets (4): Number of items part of the key.
			For a link (5): Its number of repetitions (OCCURS).
10	36		COMMENT
			Associated to the DMSII element.

Generation Elements (-GG et -DCnnnGG)

ADDITIONAL ELEMENTS OF DESCRIPTION

The definition and description lines of a Database Block provide all the logical information the System needs to generate the source program of the Block. The physical information must be indicated on lines in the 'Generation Elements' screen from the definition of a Block (-GG) or via the '-DCnnnGG' choice to modify a description line.

The user can insert comments, commands, descriptions, labels, print requests wherever he wishes to in the generated structure. He can also erase and replace the description the System generates automatically.

Comment lines can be inserted on these screens. Lines with a type '*' are used to document description lines, they are not taken into account at generation.

Several types of lines are available for the user to insert. Two types of insertions are possible:

-within an element definition lines

-within a particular item of the element

All the lines concerning the entity definition must be at the beginning, all the lines concerning an "item" must be consecutive.

Types of lines :

'V': lines generated before the automatically generated elements.

'P': lines generated between the element automatically generated parts and its description.

'Z': lines generated after the automatically generated elements.

'G': line generated instead of the automatically generated elements.

Item :

The Data Element code is indicated between the 'less than' and the 'greater than' symbols on a general documentation line of the affected entity.

<datel >

This notation is taken into account in the Data Element utilization definition.

In a dataset or a ″remap″ description, the Data Element marked this way does not interfere in the generation and must not have any line type.

For ″access″, set and subset, Data Elements marked this way are taken into account for the generation and they must have a type.

Data indication

Most often, this is performed from column two. If the data description starts in column one, a semi-colon line is inserted before the line generated for this data.

```
                                                                        *PDMCA.PDEV.HP3.851
GENERATION ELEMENTS FOR BLOCK DESC      DASDL TEST GENERATION DASDL


A LIN : T DESCRIPTION                                                          LIB
  100 : Z  % FIN DE LA DESCRIPTION DE L'ENREGISTREMENT COURS                   0851
  110 : Z  POPULATION = 1000                                                   0851
  150 : Z  VERIFY (DL10-NBHEU GTR 0 AND DL10-TCLASS LEQ 60)                    0851
  151 : Z         AND DL10-NOPROF NEQ 0                                        0851
  500 :   <IDCOUR>                                                            0874
  501 : P  REQUIRED                                                            0851
  510 :   <NBINS >                                                            0851
  511 : G  DL10-NBINS COUNT (300);                                             0851
  520 :   <SALLE >                                                            0851
  521 : P  NULL IS "NO";                                                       0851
      :
      :
      :
      :
      :
      :
      :

O: C1 CH:  -GG
```

UTILIZATION

For the Block description (-GG)

The four line types are possible.

. G / The elements automatically generated are overwritten.

. V / To indicate generation-print requests (ex:$SET ..).

. P / To enter specification parameters for instance and other Database elements.

. Z / To enter logical descriptions or any data that must be located at the end of a description.

For a description line (-DCnnnGG)

The four line types are possible.

. G / The automatically generated elements are overwritten.

. V / To indicate data before the dataset.

. P / To type lines before the dataset description (POPULATION for instance).

. Z / To enter clauses located after the dataset description (Physical options for instance).

For an ″item″

Three line types only. The notion of end (Z) is reserved to the Block and the description line.

. V / Before the generated elements concerning the item.

. G / Overwrites the generated elements concerning the item.

This utilization is useful for the codification of a virtual item with a ″field″ type.

For a group Data Element, the entire group is overwritten.

. P / After generated elements concerning the item.

This type of item will certainly be the most used as it completes an item generation with particular DMSII clauses.

When such a type of line is used, it is the user's responsability to indicate the end of instructions (semi-colon).

Details concerning the ″access″, set and subset lines.

The marked Data Element can be used to do the following:

. P / codify key items.

. Z / indicate ″DATA″ items.

This entry gives indication about the Data Element utilization. The remaining space on the line is available to enter additional information, in particular the punctuation.

Example : Generation of a KEY clause for a set on a ffss dataset:

P <datel1>,

P <datel2>DESCENDING,

P <datel3>)

generates

KEY IS (

ffss-datel1 ,

ffss-datel2 DESCENDING,

ffss-datel3)

PRESENTATION OF GENERATED ELEMENTS

For the entire block

V : -- line -GG

G : INITIALIZE;

P : -- line -GG

_ / Generated elements from the Block description.

Z : -- line -GG

For a description line

. Dataset

V : -- line -DCnnnGG

G : DATASET type name

G : ″comment″

P : -- line -DCnnnGG

G : (

_ / Generated elements from the dataset description.

G :)

Z : -- line -DCnnnGG

G : ;

. Access

V : -- line -DCnnnGG

G : ″comment″ name

G : ACCESS TO dataset

G : KEY IS (

P : -- line -DCnnnGG

Z : -- line -DCnnnGG

G : ;

. Set

V : -- line -DCnnnGG

G : ″comment″ name

G : SET OF dataset

G : KEY IS (

P : -- line -DCnnnGG

G : type

Z : -- line -DCnnnGG

G : ;

For a Data Element

. Elementary Data Element

V : -- line -DCnnnGG

G : ffss-datel type (;)

G : REQUIRED (;)

G : OCCURS n (;)

G : DEPENDING ON ffss-datel (;)

P : -- line -DCnnnGG

. Group Data Element

V : -- line -DCnnnGG

G : ffss-datel type (;)

G : REQUIRED (;)

G : OCCURS n (;)

G : DEPENDING ON ffss-datel (;)

P : -- line -DCnnnGG

G : (

G :);

NOTE: A G-type line overwrites the entire group of automatically generated lines and is located on the first line. In this case, the user must enter the right punctuation (semi-colon and brackets) using P-type or Z-type lines.

Chapter 6. Access Modes

On-line

LIST OF DATA ELEMENTS

CHOICE	SCREEN	UPD
LCEaaaaaa	List of Elements by Code (starting with Data Element 'aaaaaa').	NO
LNEaaaaaaaaaaaaa	List of Data Elements sorted by name (starting with name 'aaaaaaaaaaaaa') (case sensitive). The sort is performed on the following Elements: - the first twenty characters of the clear name, - the code of the Data Element. Note: Child Data Elements with no clear name do not appear on the list	NO
LAEaaaaaaaaaaaaa	List of Data Elements sorted by Cobol name (starting with name 'aaaaaaaaaaaaa').	NO
LREaaaaaaaaaaaaaaaaaa	List of Data Elements sorted by relational name (starting with 'aaaaaaaaaaaaaaaaaa').	NO
LFEaaaaaa	List of undefined Data Elements by code (starting with Element 'aaaaaa').	NO
LUEaaaaaa	List of Data Elements for update (starting with Element 'aaaaaa').	YES

DESCRIPTION OF DATA ELEMENT 'aaaaaa'

CHOICE	SCREEN	UPD
Eaaaaaa	Definition of Data Element 'aaaaaa'.	YES
EaaaaaaDbbb	Description of Data Element 'aaaaaa' (starting with line number 'bbb').	YES
EaaaaaaCR	Instances linked to Data Element 'aaaaaa' via User Relations.	YES
EaaaaaaGCbbb	Comments on Data Element 'aaaaaa' (starting with line number 'bbb').	YES

EaaaaaaGEbbb	Error messages on Data Element 'aaaaaa' (starting with line number 'bbb').	YES
EaaaaaaATbbbbbb	Text assigned to the Data Element 'aaaaaa' (starting with text 'bbbbbb').	NO
EaaaaaaX	X-references of Data Element 'aaaaaa' to all entities.	NO
EaaaaaaXTbbbbbb	X-references of Data Element 'aaaaaa' to texts (starting with text 'bbbbbb').	NO
EaaaaaaXMbbbbbb	X-references of Data Element 'aaaaaa' to the Method Entities (starting with Method Entity 'bbbbbb').	NO
EaaaaaaXQbbbbbb	X-references of Data Element 'aaaaaa' to instances through User Relations (starting with User Relation 'bbbbbb').	NO
EaaaaaaXBbbbbbb	X-references of Data Element 'aaaaaa' to Blocks (starting with Block 'bbbbbb').	NO
EaaaaaaXBbbbbbbDCddd	X-references of Data Element 'aaaaaa' to CODASYL-type blocks (starting with Block 'bbbbbb', line number 'ddd')	NO
EaaaaaaXBbbbbbbDHddd	X-references of Data Element 'aaaaaa' to Hierarchical-type Block (starting with Block 'bbbbbb', line number 'ddd')	NO
EaaaaaaXBbbbbbbDRddd	X-references of Data Element 'aaaaaa' to Relational-type Block (starting with Block 'bbbbbb', line number 'ddd')	NO
EaaaaaaXVbbbbbb	X-references of Data Element 'aaaaaa' to Documents (starting with Document 'bbbbbb').	NO
EaaaaaaXObbbbbb	X-references of Data Element 'aaaaaa' to Screens (starting with screen 'bbbbbb').	NO
EaaaaaaXObbbbbbWccddd	X-references of Data Element 'aaaaaa' to Work Areas (-W) of Screen 'bbbbbb' (starting with work area 'cc', line number 'ddd').	NO
EaaaaaaXObbbbbbBccddeee	X-references of Data Element 'aaaaaa' to Beginning Insertions (-B) of Screen 'bbbbbb' (starting with section 'cc', paragraph 'dd', line number 'eee').	NO

EaaaaaaXObbbbbbCPcccccc	X-references of Data Element 'aaaaaa' to Call of P.M.S.(-CP) of Screen 'bbbbbb' (starting with Macro-Structure 'cccccc').	NO
EaaaaaaXObbbbbbPccddeee	X-references of Data Element 'aaaaaa' to Procedural Code (-P) of Screen 'bbbbbb' (starting with function/subfunction 'ccdd', line number ' eee').	NO
EaaaaaaXKbbbb	X-references of Data Element 'aaaaaa' to the key of Relational /SQL Database Blocks (starting with Segment 'bbbb').	NO
EaaaaaaXSbbbb	X-references of Data Element 'aaaaaa' to Segments (starting with Segment 'bbbb').	NO
EaaaaaaXRbbb	X-references of Data Element 'aaaaaa' to Reports (starting with Report 'bbb').	NO
EaaaaaaXRbbbCE	X-references of Data Element 'aaaaaa' to Report Call of Elements (starting with Report 'bbb').	NO
EaaaaaaXPbbbbbb	X-references of Data Element 'aaaaaa' to Programs (starting with Program 'bbbbbb').	NO
EaaaaaaXPbbbbbbBccddeee	X-references of data element 'aaaaaa' to Beginning Insertions (-B) of Program 'bbbbbb' (starting with section 'cc', paragraph 'dd', line number 'eee').	NO
EaaaaaaXPbbbbbbCPcccccc	X-references of Data Element 'aaaaaa' to Call of P.M.S. (-CP) of Program 'bbbbbb' (starting with Macro-Structure 'cccccc').	NO
EaaaaaaXPbbbbbbSCfusfnnn	X-references of Data Element 'aaaaaa' to source code (-SC) of 'reversed' program 'bbbbbb' (starting with function/subfunction 'fusf', line number 'nnn')	NO
EaaaaaaXPbbbbbbWccddd	X-references of Data Element 'aaaaaa' to Work Areas (-W) of Program 'bbbbbb' (starting with Work Area 'cc', line number 'ddd')	NO
EaaaaaaXPbbbbbbPfusfnnn	X-references of Data Element to Procedural Code (-P) of Program 'bbbbbb' (starting with function/ subfunction 'fusf', line number 'nnn').	NO

```
EaaaaaaXPbbbbbb9cccccc  X-references of Data Element      NO
                        to Pure COBOL Source Code (-9)
                        of Program 'bbbbbb' (starting with
                        -9 line 'cccccc').
EaaaaaaXFbbbbbb         X-references of Data Element      NO
                        'aaaaaa' to User Entities
                        (starting with UE 'bbbbbb').
```

NOTE: After the first choice of type 'Eaaaaaa', 'Eaaaaaa' can be replaced with '-'.

All notations between parentheses are optional.

```
LIST OF SEGMENTS
----------------
CHOICE              SCREEN                              UPD
------              ------                              ---
LCSaaaa             List of Segments by code             NO
                    (starting with Segment 'aaaa').
LNSaaaa             List of Segments by name             NO
                    (starting with Segment 'aaaa')
                    (case sensitive).
DESCRIPTION OF SEGMENT 'aaaa'
-----------------------------
CHOICE              SCREEN                              UPD
------              ------                              ---
Saaaa               Definition of Segment 'aaaa'.       YES
SaaaaCR             Instances linked to Segment         YES
                    'aaaa' via User Relations.
SaaaaGCbbb          Comments on Segment 'aaaa'          YES
                    (starting with line number 'bbb').
SaaaaGEbbb          Error messages on Segment 'aaaa'    YES
                    (starting with line number 'bbb').
SaaaaGGbbb          Generation Elements for Segment     YES
                    'aaaa'(starting with line number 'bbb').
SaaaaGObbb          Generation option for Segment 'aaaa' YES
                    (starting with line number 'bbb').
SaaaaATbbbbbb       Text assigned to Segment 'aaaa'      NO
                    (starting with text 'bbbbbb').
SaaaaLSPbbbb        List of Parent Segments for Segment  NO
                    'aaaa' (starting with Parent Segment
                    'bbbb').
SaaaaLSCbbbb        List of Child Segments for Segment   NO
                    'aaaa' (starting with Child Segment
                    'bbbb').
SaaaaX              X-references of Segment 'aaaa'.      NO
SaaaaXSbbbb         X-references of Segment 'aaaa' to    NO
                    segments (starting with Segment 'bbbb').
```

SaaaaXBbbbbbb	X-references of Segment 'aaaa' to Blocks (starting with Block 'bbbbbb').	NO
SaaaaXQbbbbbb	Occurrences linked to Segment 'aaaa' through User Relations (starting with Relation 'bbbbbb').	NO
SaaaaXVbbbbbb	X-references of Segment 'aaaa' to Documents (starting with Document 'bbbbbb').	NO
SaaaaXPbbbbbb	X-references of Segment 'aaaa' to programs (starting with program 'bbbbbb').	NO
SaaaaXPbbbbbbCPcccccc	X-references of Segment 'aaaa' to Call of P.M.S. (-CP) of Program 'bbbbbb' starting with Macro-Structure 'cccccc').	NO
SaaaaXPbbbbbbWccddd	X-references of Segment 'aaaa' to Work Areas (-W) of Program 'bbbbbb' (starting with Work Area 'cc', line number 'ddd').	NO
SaaaaXObbbbbb	X-references of Segment 'aaaa' to Screens (starting with Screen 'bbbbbb').	NO
SaaaaXObbbbbbCPcccccc	X-references of Segment 'aaaa' to Call of P.M.S.(-CP) of Screen 'bbbbbb' (starting with Macro-Structure 'cccccc').	NO
SaaaaXObbbbbbWccnnn	X-references of Segment 'aaaa' to Work Areas (-W) of Screen 'bbbbbb' (starting with Work Area 'cc', line number 'nnn').	NO
SaaaaSSbn	Definition of the sub-schemas or sub-systems of Segment 'aaaa' in the Pactables function (starting with sub-schema 'n' with 'b' = 's', or sub-system 'n' with 'b' = 'y'.	YES
SaaaaCEbbb	Call of Elements/Attributes of Segment 'aaaa'(starting with line number 'bbb').	YES
SaaaaCEbbbGCccc	Comments on the Element/Attribute called on line 'bbb' of Segment 'aaaa' (starting with Comments line number 'ccc").	YES
SaaaaCEbbbGEccc	Error message on the Elem/Attribute called on line 'bbb' of Segment 'aaaa' (starting with line number 'ccc').	YES
SaaaaCEbbbGGccc	Generation Elements on the Element/ Attribute called on line 'bbb' of Segment 'aaaa' (starting with line number 'ccc').	YES

SaaaaDBEbbb	SQL view source for view 'aaaa' (starting with line 'bbb').	YES
SaaaaLALbbb	Level, address and length of Segment 'aaaa' (starting with line 'bbb').	NO
SaaaaDEDbbb	Data Element details of Segment 'aaaa' (starting with line 'bbb'). If this choice is used in C2 option, the relational label replaces that of the Data Element.	NO
SaaaaCNbbbbbb	List of constraints of Segment 'aaaa' integrity (from the block 'bbbbbb')	NO
SaaaaSTA	Statistics on Segment 'aaaa'.	NO
SaaaaACT	Activity calculation on Segment 'aaaa'.	NO

NOTE: After the first choice of type 'Saaaa', 'Saaaa' can be replaced with '-'.

All notations between parentheses are optional.

LISTS

CHOICE	SCREEN	UPD
LCBaaaaaa	List of Database Blocks by code (starting with block 'aaaaaa').	NO
LNBaaaaaa	List of Database Blocks by name (starting with block 'aaaaaa') (case sensitive).	NO
LTBaabbbbbb	List of Database Blocks by type (starting with type 'aa' and Database Block 'bbbbbb').	NO
LEBaaaaaaaa	List of Database Blocks by external name (starting with name 'aaaaaaaa').	NO

DESCRIPTION OF BLOCK 'aaaaaa'

CHOICE	SCREEN	UPD
Baaaaaa	Definition of Database Block 'aaaaaa'	YES
BaaaaaaCR	Instances linked to Database Block 'aaaaaa' through User Relations.	YES
BaaaaaaGCbbb	Comments for Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaGGbbb	Generation Elements for Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaGObbb	Generation Options for Database Block 'aaaaaa' (starting with line 'bbb').	YES

```
BaaaaaaATbbbbbb    Text Assigned to Database Block        NO
                   'aaaaaa' (starting with text 'bbbbbb').
BaaaaaaX           Cross-references of Database Block     NO
                   'aaaaaa'.
BaaaaaaXBbbbbbb    Cross-references of Database Block     NO
                   'aaaaaa' to PSB's (starting with
                   PSB 'bbbbbb').
BaaaaaaXObbbbbb    Cross-references of Database Block     NO
                   'aaaaaa' to Screens (starting with
                   Screen 'bbbbbb').
BaaaaaaXObbbbbbCScdddd
                   Cross-references of Database Block     NO
                   'aaaaaa' to the Call of Segments
                   of Screen 'bbbbbb'(starting with
                   category 'c' and with Segment 'dddd').
                   Note: 'c' is equal to & for the
                   Screen-top category.
BaaaaaaXObbbbbbWccddd
                   Cross-references of Database Block     NO
                   'aaaaaa' to the Work Areas of Screen
                   'bbbbbb' (starting with Work Area 'cc',
                   line number 'ddd').
BaaaaaaXQbbbbbb    List of occurrences linked to Database NO
                   Block 'aaaaaa' through User-Defined
                   Relation (starting with Relation
                   'bbbbbb').
BaaaaaaXVvvvvvv    Cross-references of Database Block     NO
                   'aaaaaa' to Volumes (starting with
                   Volume 'vvvvvv').
BaaaaaaXPbbbbbb    Cross-references of Database Block     NO
                   'aaaaaa' to Programs (starting with
                   Program 'bbbbbb').
BaaaaaaXPbbbbbbWccddd                                     NO
                   Cross-references of Database Block
                   'aaaaaa' to Work Areas of Program
                   'bbbbbb' (starting with Work Area 'cc',
                   line number 'ddd').
```

DESCRIPTION OF 'aaaaaa' DMSII Block

BaaaaaaDCbbb (UPD)

Description of the 'aaaaaa' block of DMSII type (from line 'bbb').

BaaaaaaDCbbbGCccc (UPD)

Comments of the line 'bbb' of the 'aaaaaa' Block (from the Comments line 'ccc').

BaaaaaaDCbbbGGccc (UPD)

Generation Elements on the line 'bbb' of the 'aaaaaa' Block (from Generation Elements line 'ccc').

NOTE: After the first choice of type 'Baaaaaa', 'Baaaaaa' can be replaced with '-'.

All notations between parentheses are optional.

BATCH

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

DATABASE BLOCK DESCRIPTION

BATCH FORM

Batch Form 'L3' is used for the description of a CODASYL, DB2, or TANDEM Database Block.

ACTION CODES

```
.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 a line.
.B      = Deletion of the data base block lines starting
          from an including the indicated line number as
          well as the associated V3 lines.
.R      = End of multiple deletion following this line.
          If no R-type line appears after a B-type
          line, the deletion ends with the last line
          number of the Block.
```

DATA ELEMENT DEFINITION

Batch Form 'C' is used for the definition of a Data Element.

DATA ELEMENT DESCRIPTION

Batch Form 'E' is used for the description of a Data Element.

SEGMENT DEFINITION

Batch Form '2' is used for the definition of a Segment.

SEGMENT DESCRIPTION

Batch Form '3' is used for the description of a Segment.

ACTION CODES

The batch action codes for these entities are identical to the ones used for the Database Block entity.

NOTE CONCERNING DELETION OF A DATA ELEMENT

Deletion of a Data Element (using ACTION CODE 'D') is only possible if the Data Element is not used in screens, reports and Segments and if it has no child Data Element.

It is possible to globally delete (using ACTION CODE 'B') a Data Element and all of its uses in screens, reports or Segments.

When a multiple deletion is done on a parent Data Element, all of its child Data Elements will be deleted along with all of the uses of the parent and child Data Elements.

Generation and/or Printing

The generation and printing of Database Blocks are requested in on-line mode on the

The following commands are available:

LTB: Lists all the Database Blocks of the Libraries of the selected sub-network, sorted by type.

- C1 OPTION: Without keywords,
- C2 OPTION: With explicit keywords.

LCB: Identical to 'LTB' but sorted by code.

LEB: Identical to 'LTB' but sorted by external name.

You can request the list of the Database Blocks which include one or more keyword(s). The corresponding command must be entered with a continuation line, on which the keywords used as selection criteria are indicated (refer to the 'Character Mode User Interface' Guide). The list is sorted by code. The corresponding command is:

LKB: Same as 'LCB' but sorted by keyword. Option 'C2' cannot be used.

DTB: Description of the Database Block whose code is in- dicated in the ENTITY field, or description of all Da- tabase Blocks if the field is not entered.

In the latter case, you can request the descriptions of all the Blocks of a given type, by spe- cifying this type in the print request.

GCB: Generation of a Database Block whose code must be indicated. Same printing option as for DTB.

Chapter 7. Example

Presentation

EXAMPLE PRESENTATION

The objective of this chapter is to present the different steps necessary to generate in DLL language.

The chapter contains the following parts:

- System screens used for the description of the DMSII Database (only the most significative screens are shown).
- Data description as the System generates it.

```
                                                                        *PDMCA.PDEV.HP3.851
DESCRIPTION OF DASDL DMSII  TDASDL TEST GENERATION DASDL


A LIN : T SET NA DATASET   OPTION     PT. COMMENT
      :   LIN IT EMB.      SET/IT     N K
  100 : 1            DL10 S      *       MAIN FILE : COURSES
  101 : 5 PROF   DL40 DL10 C
  200 : 1        DL10 DL20 U      *       BOOKS
  210 : 4 LIVK   DL10 DL20 LI     *
  300 : 1        DL10 DL30 S      *       STUDENTS
  301 : 5 SSNO   C    DL30 MFSSET
  302 : 5 SSNO1  S    DL30 MFSSET
  310 : 3 ETUSET DL10 DL30 IS     * 00002
  350 : 3 COUSET      DL10 IS     *
  400 : 1             DL40 S      *       PERSONNEL
  401 : 5 IDCOUR DL10 DL40 C
  402 : 5 TELEPH DL80 DL40 TELEPH
  403 : 5 SUPER  DL40 DL40 N
  410 : 3 SS-U-P      DL40 IS     *
  420 : 3 U-P-ST      DL40 IS     *
  500 : 1             DL50 S      *       REGISTERED
  501 : 5 TELEPH DL80 DL50 TELEPH

O: C1 CH: BDASDL DC
```

```
                                                                        *PDMCA.PDEV.HP3.851
DESCRIPTION OF DASDL DMSII  TDASDL TEST GENERATION DASDL

A LIN : T SET NA DATASET   OPTION     PT. COMMENT
      :   LIN IT EMB.      SET/IT     N K
  510 : 3 QSET   DL50 DL60 OL     *
  600 : 1        DL50 DL60 S      *       QUARTER
  610 : 3 CSEET  DL60 DL70 IS     *
  700 : 1        DL60 DL70 S      * 00002 COURSES
  710 : 1 V      DL60 DL71          00001
  720 : 1 V      DL60 DL72          00002
  721 : 5 IDCOUR DL10 DL72 C
  750 : 3 MFSSET      DL50 IS     *
  800 : 1             DL80 S      *       ADDRESS
  810 : 3 SAD         DL80 IS     *
  820 : 3 SSAD        DL80 IS     *
  830 : 4 STUAD       DL80 IS     * 00002
  840 : 4 FACAD       DL80 IR     * 00002
  850 : 4 ADMAD       DL80 IR     * 00002
  860 : 4 FREEPA      DL40 BV     *
  870 : 4 SEXSET      DL50 BV     *       STUDENTS OVER 21
  880 : 4 SMART       DL50 BV

O: C1 CH: BDASDL DC
```

```
                                                                        *PDMCA.PDEV.HP3.851
GENERATION ELEMENTS BLOCK DESC         TDASDL TEST GENERATION DASDL

A LIN : T DESCRIPTION                                                          LIB
  200 : P  <NOM   >ASCENDING,                                                  0851
  210 : P  <PRENOM>)                                                           0851
  300 : Z  DUPLICATES                                                          0851
  310 : Z  LOADFACTOR = 75 TABLESIZE = 12 AREAS = 100                          0851
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: B TDASDL DC 100 GG
```

```
------------------------------------------------------------------------------
|                                                          *PDMCA.PDEV.HP3.851 |
| GENERATION ELEMENTS BLOCK DESC        TDASDL TEST GENERATION DASDL          |

| A LIN : T DESCRIPTION                                                 LIB   |
|   010 : P  WHERE (DL80-FACETU EQL 2)                                  0851  |
|   100 : P  <ZIP   >,                                                  0851  |
|   110 : P  <SSNO  >)                                                  0851  |
|   200 : Z  DUPLICATES MODULUS = 97                                    0851  |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |

|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|                                                                             |
| O: C1 CH: B TDASDL DC 101 GG                                                |
------------------------------------------------------------------------------
```

Description of DASDL Generated Elements

```
 % DASDL GENERATION EXAMPLE
 $ SET LIST SINGLE STORE TEST
INITIALIZE;
DL10  STANDARD   DATASET
"MAIN FILE : COURSES          "
    (
    DL10-IDCOUR    GROUP
    REQUIRED
        (
        DL10-DEPART    ALPHA(2);
        DL10-NIVEAU    NUMBER(3);
        DL10-COURSN    NUMBER(4);
        );
    DL10-NOPROF    NUMBER(2);
    DL10-NBINS COUNT (300);
    DL10-SEMAIN    FIELD
        (
        DL10-LUNDI     BOOLEAN;
        DL10-MARDI     BOOLEAN;
        DL10-MERCDI    BOOLEAN;
        DL10-JEUDI     BOOLEAN;
        DL10-VENDDI    BOOLEAN;
        DL10-SAMEDI    BOOLEAN;
        );
```

```
DL10-IMMEU     NUMBER(3);
DL10-SALLE     ALPHA(2)
NULL IS "NO";
DL10-COURS     ALPHA(24);
DL10-FLAGS     FIELD(12);
DL10-NBHEU     NUMBER(4);
DL10-TCLASS    NUMBER(2);
DL10-PROF
               IS IN  DL40  COUNTED
               OCCURS 3 TIMES;
DL20  UNORDERED  DATASET
"BOOKS                                "
    (
    DL20-IDLI      NUMBER(9);
    DL20-TITRE     ALPHA(60)
    NULL IS BLANKS;
    DL20-AUTEUR    ALPHA(30);
    )
    BUFFERS = 1 + 1 PER USER,
    AREAS = 10,
    AREASIZE = 500,
    POPULATION = 5,
    BLOCKSIZE = 5
;
LIVK
    SUBSET OF  DL20
    UNORDERED LIST
    DATA
    DL20-IDLI
;
DL30  STANDARD   DATASET
"STUDENTS                             "
    (
    DL30-NOM       ALPHA(15)
                   REQUIRED;
    DL30-PRENOM    ALPHA(10)
                   REQUIRED;
    DL30-SSNO
                   IS IN     MFSSET;
    DL30-SSNO1
                   IS KEY OF MFSSET;
    )
    POPULATION = 300
;
ETUSET
    SET    OF  DL30
    KEY IS (
    DL30-NOM    ASCENDING,
    DL30-PRENOM )
    INDEX SEQUENTIAL
    DUPLICATES
    LOADFACTOR = 75 TABLESIZE = 12 AREAS = 100
;
)
% RECORD COURSES : END OF DESCRIPTION
```

```
    POPULATION = 1000
    VERIFY (DL10-NBHEU GTR 0 AND DL10-TCLASS LEQ 60)
    AND DL10-NOPROF NEQ 0
;
COUSET
    SET    OF  DL10
    KEY IS
    DL10-IDCOUR DESCENDING
    INDEX SEQUENTIAL
    NO DUPLICATES
;
DL40  STANDARD   DATASET
"PERSONNEL                          "
    POPULATION = 997
    (
    DL40-NBPER COUNT (100);
    DL40-NOMC      GROUP
    REQUIRED
        (
        DL40-NOM       ALPHA(15);
        DL40-PRENOM    ALPHA(10);
        );
    DL40-SEXE      BOOLEAN;
    DL40-AGE       NUMBER(2)
    NULL IS HIGH-VALUE;
    DL40-SSNO      NUMBER(9)
                   REQUIRED;
    DL40-DPT       ALPHA(4);
    DL40-RANG      ALPHA(1);
    DL40-SALAIR    NUMBER(S7,2)
    INITIALVALUE IS LOW-VALUE;
    DL40-IDCOUR
                   IS IN  DL10  COUNTED
                   OCCURS 8 TIMES;
    DL40-TELEPH
                   IS IN  DL80  VERIFY ON  DL80-TELEPH;
    DL40-SUPER
                   IS IN  DL40  WITH NO PROTECTION;
    )
;
SS-U-P
    SET    OF  DL40
    KEY IS
    DL40-SSNO
    INDEX SEQUENTIAL
    NO DUPLICATES
;
U-P-ST
    SET    OF  DL40
    KEY IS
    DL40-NOMC
    INDEX SEQUENTIAL
    DUPLICATES
;
DL50  STANDARD   DATASET
```

```
"REGISTERED                          "
    (
    DL50-SSNO      NUMBER(9)
                   REQUIRED;
    DL50-NONOM     NUMBER(1);
    DL50-LNOM      ALPHA(30);
    DL50-ALIAS     ALPHA(30)
                   OCCURS 9;
    DL50-FNOM      ALPHA(30);
    DL50-ADRCAM    GROUP
        (
        DL50-DORTOI    ALPHA(6);
        DL50-ROOM      NUMBER(4);
        DL50-BOXE      NUMBER(4);
        DL50-POSTE     NUMBER(7);
        );
    DL50-ND        NUMBER(2);
    DL50-DEGRE     ALPHA(4)
                   OCCURS 6;
    DL50-TOTHEU    NUMBER(3);
    DL50-TOTQP     REAL(3);
    DL50-MPTGRA    NUMBER(3,2);
    DL50-MJR       NUMBER(3);
    DL50-AMJR      ALPHA(18);
    DL50-SEXE      BOOLEAN;
    DL50-AGE       NUMBER(2);
    DL50-TELEPH
                   IS IN  DL80  VERIFY ON  DL80-TELEPH;
    QSET
        SET    OF  DL60
        KEY IS
        DL60-QTIER
        NO DUPLICATES
        ORDERED LIST
    ;
    DL60  STANDARD   DATASET
    "QUARTER                             "
        (
        DL60-QTIER     ALPHA(4)
                       REQUIRED;
        DL60-QTTHRS    NUMBER(2);
        DL60-QTRQP     NUMBER(2);
        CSEET
            SET    OF  DL70
            KEY IS
            DL70-TYCOUR
            INDEX SEQUENTIAL
            DUPLICATES
        ;
            POPCORSES POPULATION (100000) OF DL70;
        DL70  STANDARD   DATASET
        "COURSES                               "
            POPULATION = 4, BLOCKSIZE = 30 WORDS
            (
            DL70-TYCOUR    NUMBER(1)
```

```
                        REQUIRED;
            DL70-CORTYP  RECORD TYPE(2);
            )
            ,
        1:
            (
            DL71-GRADE     ALPHA(2);
            DL71-IDCOUR    ALPHA(9);
            )
            ,
        2:
            (
            DL72-YR        NUMBER(2);
            DL72-COEFF     NUMBER(2);
            DL72-IDCOUR
                           IS IN  DL10  COUNTED;
            DL72-GCD       ALPHA(2);
            DL72-DIPLO     ALPHA(30);
            DL72-PPGRD     ALPHA(2);
            )
        ;
        )
        % "QUARTER" : END OF DESCRIPTION
    ;
    )
    % "REGISTERED" : END OF DESCRIPTION
    LOCK TO MODIFY DETAILS       % PHYSICAL OPTION
    BLOCKSIZE = 6
    POPULATION = 5000
    AREAS = 100
    KIND = DISK
;
MFSSET
    SET    OF  DL50
    KEY IS
    DL50-SSNO
    INDEX SEQUENTIAL
    NO DUPLICATES
;
DL80  STANDARD   DATASET
"ADDRESS                              "
    (
    DL80-FACETU    NUMBER(1);
    DL80-SSNO      NUMBER(9)
                   REQUIRED;
    DL80-NUMLNS    NUMBER(1);
    DL80-ADRLN     ALPHA(30)
                   OCCURS 9;
    DL80-ZIP       NUMBER(5)
                   REQUIRED;
    DL80-TELEPH    NUMBER(7);
    )
    ;
    POPADMAD POPULATION (14) OF ADMAD
;
```

```
SAD
    SET    OF  DL80
    KEY IS
    DL80-ZIP
    INDEX SEQUENTIAL
    DUPLICATES FIRST
;
SSAD
    SET    OF  DL80
    KEY IS
    DL80-SSNO
    INDEX SEQUENTIAL
    DUPLICATES LAST
;
STUAD
    SUBSET OF  DL80
    WHERE (DL80-FACETU EQL 1)
    KEY IS (
    DL80-ZIP    ,
    DL80-SSNO   )
    INDEX SEQUENTIAL
    DUPLICATES
;
FACAD
    SUBSET OF  DL80
    WHERE (DL80-FACETU EQL 2)
    KEY IS (
    DL80-ZIP    ,
    DL80-SSNO   )
    INDEX RANDOM
    DUPLICATES MODULUS = 97
;
ADMAD
    SUBSET OF  DL80
    WHERE (DL80-FACETU EQL 3)
    KEY IS (
    DL80-ZIP    ,
    DL80-SSNO   )
    INDEX RANDOM
    DUPLICATES
;
FREEPA
    SUBSET OF  DL40
    WHERE (DL40-SALAIR LSS 0 OR DL40-SALAIR EQL 0)
    BIT VECTOR
;
SEXSET             "STUDENTS OVER 21                    "
    SUBSET OF  DL50
    WHERE (DL50-AGE GEQ 21 AND NOT DL50-SEXE)
    BIT VECTOR
;
SMART
    SUBSET OF  DL50
    BIT VECTOR
;
```

```
DMUTIL
    SUBSET OF  DL50
    BIT VECTOR
;
 %%% END OF EXAMPLE
```

Part Number: DDDM2000351A - 7658

Printed in USA