

VisualAge Pacbase



Data Dictionary

Version 3.5



VisualAge Pacbase



Data Dictionary

Version 3.5

Note

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

According to your licence agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

<http://www.ibm.com/software/awdtools/vapacbase/productinfo.htm>

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

First Edition (May 2004)

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.htm> or to the following postal address:

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

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

© Copyright International Business Machines Corporation 1983,2004. All rights reserved.

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

Contents

Notices	v	Segments: On-Line Access	63
Trademarks.	vii	Segments: Generation-Print	71
Chapter 1. Introduction	1	Chapter 5. Database Blocks.	73
Purpose of the Manual	1	Database Blocks: Introduction	73
Entities Managed by the System	1	Database Blocks: Definition	73
Principles of Description	2	Hierarchical Database Blocks: Description	78
Chapter 2. Data Elements	3	Relational/SQL Database Blocks: Description	80
Data Elements: Introduction	3	CODASYL, TANDEM And DB2 Blocks: Description	84
Data Elements: Definition	4	Database Blocks: On-Line Access	90
List of Data Elements for Update	13	Database Blocks: Generation-Print	97
Data Elements: Description	14	Chapter 6. Common Screens	99
Data Elements: On-Line Access	21	Introduction	99
Data Elements: Generation-Print	33	Error Messages - Help (-GE).	99
Chapter 3. Data Structures	35	Generation Elements (-GG).	103
Data Structures: Introduction	35	Generation Options (-GO)	106
Data Structures: Definition	36	Chapter 7. Generation of Copy Books 111	
Data Structures: On-Line Access	38	Principles.	111
Data Structures: Generation-Print	45	Description of 'DATA' PIA	111
Chapter 4. Segments.	47	Description of 'DATASQ' PIA	114
Segments: Introduction	47	Data Generation and/or Printing	116
Segments: Definition	47	Example of Generated 'DATA' in the FILE SECTION.	117
Segments: On-Line Definition	49	Example of Generated 'DATA' in WORKING-STORAGE	118
Segments: Description	53		

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

Purpose of the Manual

This Manual presents all the entities, managed by the Specifications Dictionary Function, which describe the data used to build applications.

These entities are presented here in their character-mode display but you can also access them via the Developer workbench and via the VisualAge Pacbase WorkStation.

Before reading this Manual, you should be familiar with:

- The principles described in the 'VisualAge Pacbase Metamodel and Repository' Manual. These principles are supposed to be known here.
- All the possible operations on entities. These operations are detailed in the VisualAge Pacbase WorkStation User Interface Guide (formerly VisualAge Pacbase WorkStation Reference Manual) or in the Character-Mode User Interface Guide, depending on your interface.

(The operations performed with the Developer workbench are described in the Developer workbench on-line help).

Entities Managed by the System

The entities which describe data are the following ones:

- Data Elements,
- Data Structures,
- Segments,
- Database Blocks.

Moreover the Dictionary Function manages entities which are used to document applications:

- Texts, documented in the VA Pac WorkStation User Interface Guide or in the Character-Mode User Interface Guide, depending on your interface.
- Input Aids, documented in the Developer workbench or the Character-Mode User Interface Guide, depending on your interface.
- Keywords, documented in the Developer workbench or the Character-Mode User Interface Guide, depending on your interface.

DATA ELEMENTS

Data Elements (entity 'E') are used to create a dictionary of basic units of data. This entity is made up of a Definition screen and one or more Description screens. It may be commented and may include indications for the generation of error messages.

DATA STRUCTURES

The purpose of Data Structures (entity 'D') is to group Segments together in a logical manner.

This entity is made up of a Definition screen. It may be commented, may include generation elements and options. It may also include indications for the generation of on-line help.

SEGMENTS

Segments (entity 'S') are structured sets of Data Elements.

This entity is made of a Definition screen and one or more Description screens (Call of Elements). Both the Definition and each Description line may be commented.

Other pieces of information may be added, according to the future use of the segment (file record, database segment, table item, work area....).

A Segment may include generation options/and elements and include indications for the generation of error messages and on-line help.

DATABASE BLOCKS

Database Blocks (entity 'B') describe the relations between defined Segments. These relations can be structured hierarchically, in a network, or relationally.

This entity is made up of a Definition screen and one or more Description screens. Each can be commented.

Database Blocks are described according to types. One type of Description screen is used for hierarchical Blocks, a second type is used for Blocks organized relationally, and another for Blocks organized in networks.

Database Blocks can contain generation elements and/or options.

Principles of Description

This Manual describes some of the entities managed by VisualAge Pacbase. For each, you will find;

- An introduction which explains the purpose and the general characteristics of the entity,
- A detailed description of all the entity's input fields. These fields are assigned order numbers in the screen map. These numbers are also used in the screen description that follows.

>>>: If you use Developer workbench, refer to its on-line help.

Note: If you use the VisualAge Pacbase WorkStation, the graphical interface of the corresponding windows is described in the WorkStation User Interface Guide (formerly WorkStation Reference Manual).

Chapter 2. Data Elements

Data Elements: Introduction

The purpose of the Data Element entity is to identify and describe all pieces of information used by the applications of the company.

A Data Element is a unit of data, considered 'elementary' during at least one stage in the development of a project.

For example, during functional analysis, the user keeps track of a piece of information 'DATE OF MARRIAGE', which could, during implementation, be broken down into year, month and day of marriage.

Each one of these four basic units of data is defined as a Data Element:

.DATE OF MARRIAGE,

.YEAR OF MARRIAGE,

.MONTH OF MARRIAGE,

.DAY OF MARRIAGE.

GENERAL CHARACTERISTICS

The Data Element includes the following:

- A Definition screen, (required), for input of its general characteristics (name, formats, keywords, etc.);
- A Description screen, (optional), for input of all valid values, as well as labels which can be used by the Pacbench C/S, On-Line Systems Development, Pactables and DBD Functions;
- A Comments screen (see the 'Character Mode User Interface Guide', chapter 'Documentary Facilities').
- An Error Messages - Help screen (see chapter 'Common Screens' in this Manual).

RESULTS

Once defined, Data Elements appear in:

- Lists sorted by code or name,
- Cross-references to the Programs, Screens, Reports, Segments and Volumes in which they are used,
- Volumes for end-users; their descriptions are then adapted to the targeted readership.

Data Elements: Definition

A Data Element is defined by a mnemonic code, a name and three formats (input, internal and output, which are expressed in COBOL). Whenever the Data Element is used, it automatically takes on one of these three formats.

NOTE:

If you create a Data Element by duplication (code overwrite) and if the new code already exists in a lateral library, 'W' is displayed in the ACTION CODE. It inhibits the immediate update. To perform the actual update, press the ENTER key again.

PARENT DATA ELEMENT

Technical variants of Data Elements, on which the data element's characteristics are modified, may be created.

Each variant is defined as a 'CHILD' Data Element of the 'PARENT' Data Element. The characteristics of the parent may be modified on each child Data Element.

All Data Elements from the same 'family' are logically linked in the Database.

A child Element cannot be a parent Element as well.

NOTES:

On a list screen with an operation other than 'C1', (i.e. consultation of other than the selected library and all higher level libraries), the information concerning child Data Elements is not displayed unless it is different from that of the parent Data Element.

In a child Data Element Definition screen, data different from the parent Element are identified by an asterisk (*).

If the parent Element has description lines, it is indicated in the child Element Description screen by a special line marked by an asterisk (*) in its ACTION CODE field, and labeled "PARENT ELEMENT:". These description lines can be visualized with the value 'C2' entered in the OPERATION CODE field of the child Element's Description screen.

When a parent Data Element is consulted, the first 40 child Data Elements are also listed.

If there are more than 40 child Data Elements, the 40th line will display the value '*MORE.'. In this case, all child Data Elements may be viewed on the screens listing Data Elements by code and by name, 'LCE' and 'LNE' screens, respectively.

DATE PROCESSING

Data Elements used to represent dates are automatically managed by the system. A symbolic format can be assigned to these Data Elements, as follows:

- 'D' for a display type format (input):
X(6) (MMDDYY or DDMMYY according to the value of the DATE FORMAT IN GENERATED PROGRAMS entered on the Library Definition screen),
- 'I' for an internal type format:

- X(6) (YYMMDD),
- 'E' for an output format (extended):
X(8) (MM/DD/YY or DD/MM/YY according to the value of the DATE FORMAT IN GENERATED PROGRAMS entered on the Library Definition screen),
- 'C' for a display type format with century (input):
X(8) (MMDDCCYY or DDMMCCYY according to the value of the DATE FORMAT IN GENERATED PROGRAMS entered on the Library Definition screen),
- 'S' for internal type format with century:
X(8) (CCYYMMDD),
- 'M' for an output format with century (extended):
X(10) (MM/DD/CCYY or DD/MM/CCYY according to the value of the DATE FORMAT IN GENERATED PROGRAMS entered on the Library Definition screen).
- 'G' for Gregorian format with century:
X(10) (CCYY-MM-DD),

When using the On-line Systems Development or Pacbench C/S Function, date validation is automatically performed.

NUMERIC FIELDS PROCESSING

To enter a numeric Data Element format longer than 10 characters, omit the '9' that would normally be entered after the 'V'.

EXAMPLE: S9(10)V9(3) must be entered as S9(10)V(3).

This way of coding must not be used when the format is shorter than 10 characters.

AUTOMATIC CONVERSION OF INTERNAL USAGE

The USAGE clause of a COBOL numeric variable allows you to indicate the internal representation of its value. Different USAGES are available depending on the COBOL variants adapted to the different materials.

The INTERNAL USAGE characteristic of a Data Element corresponds to the COBOL USAGE clause. You should choose a Data Element INTERNAL USAGE according to the following elements:

- The type of COBOL to generate associated with the library where you define the Data Element.
- The internal representation you want.
For example, if you generate for IBM, C INTERNAL USAGE generates USAGE COMP and F generates USAGE COMP-1. For UNISYS 1100, H INTERNAL USAGE generates USAGE COMP.

You can use this Data Element in a lower level library whose type of COBOL to generate is different to the one of the higher library.

For example, you have defined the DELCO Data Element in the HIG library with C as INTERNAL USAGE and you use it in a LOW library with a type of COBOL to generate for UNISYS 1100.

When this happens, the System automatically replaces the Data Element INTERNAL USAGE with an equivalent which is compatible with the type of COBOL to generate.

So, when you visualize DELCO in LOW, the INTERNAL USAGE automatically displayed is H. This value will be used to generate.

If there is no equivalent, or if the provided one does not suit you, you can modify the INTERNAL USAGE of the Data Element in the lower library to obtain the result you want.

UNDEFINED DATA ELEMENTS

It is possible in limited occasions to use element codes which are not defined in the dictionary; for example groups with no functional meaning.

Cross-references and a specific list are also available for these Data Elements, so that the dictionary administrator is able to control their use.

ASSOCIATED LINES

Comments (-GC).

This screen enables you to insert additional explanatory text.

'Error messages - help' (-GE).

This screen enables you to describe error messages and help texts on the Data Element.


```

-----
PURCHASING MANAGEMENT SYSTEM          SG000008.LILI.CIV.1583
DATA ELEMENT CODE    1 CITY

NAME.....: 2 CITY
TYPE.....: 3 R

INPUT FORMAT.....: 4 X(15)           LENGTH...:
INTERNAL FORMAT....: 5 X(15)           USAGE : 6 D  LENGTH...:
OUTPUT FORMAT.....: 7 X(15)           Z: 8      LENGTH...:

EXPLICIT KEYWORDS: 9

PARENT ELEMENT.....: 10

SESSION NUMBER.....: 0059           LIBRARY.....: CIV   LOCK....:

O: C1 CH: Ecity                       ACTION:
-----

```

NUMLEN	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.
		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:

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	'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.
	'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.
4	10	Input format
		(Default option: INTERNAL FORMAT).
		This format is normally used with input transaction files (the conversion to internal format is done when updating master files).
		The user must ensure the compatibility between input and internal formats.
		The input format will automatically be used in segment descriptions.
		For batch programs, the user may select the format on the Call of Data Structures (-CD) screen.
		The input format must be coded like a COBOL picture (without print characters).
		USAGE is always DISPLAY.
		This format is not necessary for a property.
		For data elements representing a date, it is possible to assign a symbolic format:
		Display type formats (input):
	D	Without century (DDMMYY or MMDDYY)

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	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)
	TS	TIMESTAMP format.
		PACMODEL function: this field may be omitted for a property.
		For a complete list of the uses of formats with the various Database Block types, see the summary tables in Chapter "COLUMN: DATA ELEMENT" in the RELATIONAL/ SQL DATABASE DESCRIPTION reference manual.
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).
	TS	TIMESTAMP format

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			METHODOLOGY function: This field may be left blank 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" 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.
		'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.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	'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.
	'T'	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.
	'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
		(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 can also be used in a segment description.
		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 (DDMMYY or MMDDYY)
	C	With century (DDMMCCYY or MMDDCCYY)
		Internal type formats:
	I	Without century (YYMMDD)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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)
		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" Manual.
8	1		BLANK WHEN ZERO CLAUSE
			This field does not exist in batch mode: the USAGE field must be used instead.
			With OUTPUT FORMAT only:
		'Z'	Generates the 'BLANK WHEN ZERO' clause for a Data Element used in a Batch Program only.
			(For the generation of this clause with the O.L.S.D. and Pacbench C/S Functions, refer to the Data Element Description screen (-D).)
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.

List of Data Elements for Update

Data elements are the basic units of data managed by the system.

The initial building phase of the Database, or the massive input of new Data Elements, calls for a system facility which allows quick execution:

For this purpose, Data Elements may also be updated in a list format.

HOW IT WORKS

The screen used for Data Element input by list is obtained by the CHOICE 'LUE'. Depending on which code is entered in the OPERATION field, the 'LUE' screen allows input of:

- . 'C1': the basic characteristics of the Data Elements (parent, name, internal format, usage),
- . 'C2': the input and output formats.

The information to be entered here is the same as that found on the Definition Screen and thus undergoes the same validation operations.

Each line entered will appear as a Data Element Definition (E) screen.

NOTE: This screen does not allow for the specification of EXPLICIT KEYWORDS. These must be added as needed, on each individual Data Element Definition screen ('E.....').

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583	
LIST OF DATA ELEMENTS FOR UPDATE			
A ELEM.	PARENT TYPE	NAME OF DATA ELEMENT	US INT FORM LIBR.
ACCTNO :	R	ACCOUNT NUMBER	3 9(8) 0059
ACTION :	R	TRANSACTION CODE	D X 0059
. APPLI :	R	PACBASE APPLICATION CODE	D X(03) *CEN
CHOICE :	R	OPERATION CODE	D X 0059
CITY :	R	CITY	D X(15) 0059
. CLELE :	R	PACBASE ERROR MESSAGE KEY	D X(17) *CEN
DATE :	R	STANDARD DATE	D X(6) 0059
ERMSG :	R	PACBASE ERROR MESSAGE LABEL	D X(66) 0093
. ERUT :	R	ERUT VECTOR	D X *CEN
. GRAER :	R	PACBASE ERROR MESSAGE GRAVITY	D X *CEN
ITPRIC :	R	ITEM TOTAL PRICE	3 9(9)V99 0059
ITBALN :	R	ITEM ACCOUNT BALANCE	3 S9(10)V(3) 0059
ITQORD :	R	ITEM QUANTITY ORDERED	3 9(6) 0059
ITQREC :	R	ITEM QUANTITY RECEIVED	3 9(6) 0059
. LIERR :	R	PACBASE ERROR MESSAGE LABEL	D X(66) *CEN
. LINE :	R	PRINT LINE FOR BATCH REPORTS	D X(132) *CEN
. NUERR :	R	PACBASE ERROR MESSAGE NUMBER	D X(03) *CEN
. NUERR9 :	R	PACBASE ERROR MESSAGE NUMBER	D 9(03) *CEN
. NULIG :	R	PACBASE LINE NUMBER	D 9(03) *CEN

O: C1 CH: LUE

Data Elements: Description

This screen is used to provide the full description of a Data Element by assigning explanatory text as well as values or ranges of possible values and their meaning.

It is also used to complement the description by providing specific information which will be exploited where the Data Element is used (screen labels and format, relational databases name...).

GENERAL CHARACTERISTICS

Each value or range of values is entered as a code and a label; this label will be used in user documentation. The values or ranges may be used in the Data Element validation process, if desired, in batch and on-line programs.

The description lines of the parent Data Element automatically attributed to its child Data Elements. These lines are identified by an asterisk (*) in the ACTION CODE field, and may be viewed in option C2.

They cannot be modified or deleted at the level of the child Data Element.

Description lines specific to the child Data Element are displayed following those of the parent Data Element.

It is not necessary to create blank lines, as the SKIP OR ACTION TYPE field provides for line or page skip, both being taken into account in user manuals and volumes.

CHARACTERISTICS SPECIFIC TO ON-LINE SYSTEMS DEVELOPMENT and PACBENCH C/S

When using a Data Element in a Screen, a Short Label, a Column Label, a sample value and possibly an on-line format can be defined at this level, if necessary.

Delimiters of Data Element labels may be parameterized. The delimiter default value may then be overridden if it is used in the given label. A delimiter is required for the column label but optional for the short label.

Conversational Formats: In on-line programs, the format used for unprotected (variable) Data Elements is calculated by the system, as an expanded version of the INTERNAL FORMAT. For protected fields, the system will use the OUTPUT FORMAT. However you may override these formats by specifying a conversational format.

For dates, the symbolic date formats may be used.

Operation and Action codes: their values and the corresponding internal values may be indicated here.

PREREQUISITE

The Data Element must have been previously defined.

OPERATION FIELD

C1: default value.

C2: displays the source of lines entered.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		One of the values entered can be referenced as the default value. When the value 'D' is entered on the Segment Call of Elements (-CE) screen in the TYPE : VALIDATION, UPDATE, VALUES field, this value is assigned as the initial value.
		SPECIAL TYPES (OLSD, Pacbench C/S, Pactables Functions)
	'P'	DATA ELEMENT PRESENTATION VALUE:
		The sample value is entered in the SIGNIFICANCE - DESCRIPTION field. This value is used when simulating a screen for documentary purposes.
	'L'	DATA ELEMENT SHORT LABEL: Maximum length: 18 characters. NOTE: This length may be shortened by explicitly entering a delimiter (see description of the DATA ELEMENT VALUE field). Default delimiter is 'E'.
	'C'	COLUMN LABEL:
		The Column Label is defined on a single line but may use up to three lines. A delimiter in the Column Label indicates a line skip. The Column Label length is that of its longest line. Maximum length = 18 characters, including delimiters. A Column Label must be delimited by at least one delimiter (default = '/'). NOTE: To change the default delimiter, enter its value left-justified in the DATA ELEMENT VALUE field (refer to the description of this field).
	F	CONVERSATIONAL FORMAT: Data Elements used in input and output on-line:
		For Date Data Elements, enter the one-character symbolic value that represents the desired format, in the DATA ELEMENT VALUE field. The system will display the format in the SIGNIFICANCE - DESCRIPTION field.
		For other Data Elements, enter the desired output format in the SIGNIFICANCE - DESCRIPTION field.
		For numeric Data Elements, a BLANK WHEN ZERO clause may be obtained by entering 'Z' following the format entered in the SIGNIFICANCE - DESCRIPTION field.
		T ... SIGNIFICANCE - DESCRIPTION
		F ... 9(4) Z
	'O'	Declaration of the OPERATION CODE values.
	'I'	Declaration of the ACTION CODE values.
		For values 'O' and 'I', see also the SKIP OR ACTION TYPE field, and refer to the 'On-Line Systems Development' Manual and to the 'Pacbench C/S: Business Logic and TUI Clients' Manual (chapter TUI Client).
		RELATIONAL DATABASES:
	R	This value generates the Data Element's relational name on 18 characters, which is entered in the SIGNIFICANCE - DESCRIPTION field.
		The relational name of a parent Data Element is not carried forward to the child Data Element.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		With TurboImage, this field generates an Item name different from the Data Element code. In this case only the first 16 characters are recognized.
	E	This value allows you to input non standard date format in the SIGNIFICANCE - DESCRIPTION field.
		The format indicated on the Data Element Definition screen must be X(n), with n < 28 (or n < 15 for an ORACLE Database for the automatic management of dates in ON-LINE SYSTEMS DEVELOPMENT and PACBENCH C/S).
		This format is taken into account: . in the SQL generation to generate DATE for ORACLE, SYBASE and SQL SERVER, and DATETIME for INFORMIX, NONSTOP SQL. . in the OLSD and Pacbench C/S generation for the SQL accesses (e.g. by generating the TOCHAR and TODATE functions for ORACLE). Non-standard dates are not controlled in the generated programs; only standard dates (types C, D, E, G, I, M, S) are controlled. Furthermore, the date operator (AD) cannot be applied to this non-standard format.
		The system controls only the elements of the format, and not the way you put them together (ex: MD will be rejected but MMMMMM and YY-DD/MM will be accepted).
		DATA ELEMENTS COMING FROM REVERSE ENGINEERING:
	S	The COBOL data-name(s) of the associated REVERSE Elements are generated in the SIGNIFICANCE - DESCRIPTION field.
		COBOL COPYBOOKS:
	A	For COPYBOOKS, when a variant Data Element is being used as an alias-type Element, the SIGNIFICANCE - DESCRIPTION field contains the SEGMENT CODE of the Segment in which the parent is called.
		ADABAS DATABASE:
	A	For a Data Element used in an Adabas Database. This enables you to enter the values for the generation of the Format-Buffer.
		LIST OF TURBOIMAGE CLASSES:
	T	Values of the TurboImage class list.
5	1	ACTION TYPE, JUMP, CONTINUATION
		This field is used to specify:
		Line skip or page skip (only taken into account when printing User Manuals and Volumes).
		Continuation of a value range when a value does not fit on a single line.
		Operation or Action Code (also see the TYPE OF LINE field).
		SKIP:
	blank or 1	New line.
	2	1 blank line + 1 new line.
	3 to 9	2 to 8 blank lines + 1 new line.
	*	Only in User Manuals ('U' entity) : Page skip.
		CONTINUATION OF A VALUE RANGE:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'+'	This value indicates a continuation of a value range if it cannot fit on a single line.
			ACTION (OLSD function and Pacbench C/S TUI Clients)
			Two categories of value according to the selected TYPE OF LINE:
			WITH TYPE OF LINE 'T':
		'C'	Creation.
		'M'	Modification.
		'D'	Deletion.
		'X'	Mod-4 (implicit update).
			WITH TYPE OF LINE 'O':
		'A'	Display.
		'M'	Update.
		'S'	Next screen.
		'E'	End of session.
		'P'	Same page.
		'O'	Call of another screen.
6	10		DATA ELEMENT VALUE
			This field is used to specify the authorized values of the data element.
			These values undergo automatic validation if they are entered as either numeric or alphanumeric literals (quotes for the latter),
			If the Data Element takes on a range of values, the range must be described as two values between parentheses and separated by at least a space. Inverted parentheses indicate that the given value is excluded from the range.
			EXAMPLES:
			('E' 'Z') : from E inclusive to Z inclusive,)0 100(: from 0 exclusive to 100 exclusive.
			If the description of a value calls for several lines, the value must be entered on the first line.
			The values assigned to a parent Data Elements are automatically assigned to each one of its child Data Elements.
			OLSD FUNCTION & PACBENCH C/S
		'*9'	Numeric Data Element. This causes a COBOL NOT NUMERIC check to be generated.
		'*B'	Numeric Data Element: LEADING blanks are replaced by zeros.
		'*Z'	Numeric Data Element: ALL blanks are replaced by zeros.
		'*A'	Alphabetic Data Element: checks that all characters are alphabetic.
		'*L'	Alphabetic Data Element: checks that all characters are lowercase alphabetic..
		'*U'	Alphabetic Data Element: checks that all characters are uppercase alphabetic.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The system displays a decoded representation, in the SIGNIFICANCE - DESCRIPTION field.
		WITH TYPE OF LINE = 'F'
	I	Without century (picture x(6)): YYMMDD
	S	With century (picture x(8)): CCYYMMDD
	D	Without century (picture x(6)): MMDDYY or DDMMYY depending on the value entered in the DATE FORMAT IN GENERATED PROGRAMS field on the Library Def. screen.
	C	With century (picture x(8)): MMDDCCYY or DDMMCCYY depending on the value entered in the DATE FORMAT IN GENERATED PROGRAMS field on the Library Def. screen.
	G	With century (picture x(10)): CCYY-MM-DD in a Gregorian format.
		Date with slashes:
	E	Without century (picture x(8)): MM/DD/YY or DD/MM/YY.
	M	With century (picture x(10)): MM/DD/CCYY or DD/MM/CCYY
		WITH TYPE OF LINE = 'C':
		Enter the delimiter for the end of each Column label line (left-justified). Default value is '/'. WITH TYPE OF LINE = 'L':
		Enter the delimiter for the end of the short label, (left-justified). Default value is '£'. WITH TYPE OF LINE = 'O' OR 'I':
		When setting the value of the Operation and/or Action Codes via an element on the screen, enter the value that corresponds to the specific operation or action. NOTE: These values correspond to the internal operation and action codes as entered in the SKIP OR ACTION TYPE field.
	T	Time.
	TS	Timestamp.
		Concerning 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.
7	54	SIGNIFICANCE - DESCRIPTION
		The value entered here depends upon the value of the TYPE OF LINE field.
		With ' ', 'D', 'O', 'I': Enter a descriptive comment (optional).
		With 'L', 'C', or 'P': Enter the label (with delimiters as needed) or a presentation value.
		With 'A': Enter the SEGMENT CODE where the parent Data Element is called.
		With 'R': Enter the Relational Column name.
		With 'E': Enter the non-standard date format with one or several of the following elements:

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		. YY : year (YYYY with the century)
		. MM : month
		. MON : month's 3 first characters
		. DD : day
		. HH : hour 00 to 23 save for SQL Oracle : 00 to 12
		. HHAM or HHPM: hour 00 to 12 + am/pm indicator
		. HH24 : hour (00 to 23) for SQL Oracle
		. MI : minute
		. SS : second
		. FF : millisecond
		. delimiters / . : - blank
		For more information, refer to the DBMS documentation. For NONSTOP SQL: input of start field and end field.
		With 'F' (for Data Elements other than dates): Enter the output format (using standard COBOL syntax). Note: To generate a BLANK WHEN ZERO clause with numeric Data Elements, follow the format with a blank and a 'Z' (Example: 9(4) Z).
	\$OFF \$ON	When the Data Element Description is to be printed in a Document (with print option EO), the left-justified \$OFF command allows you to exclude from this printing the following Description lines. The explicit exclusion end command is \$ON, also left-justified, to be entered just after the last line to exclude from printing. WARNING: This exclusion is not effective when the Data Element Description lines appear in a generated online help. Only lines bearing the \$OFF and \$ON commands are excluded. For more information about the \$OFF and \$ON commands, refer to the "Personalized Documentation Manager" Manual.

Data Elements: On-Line Access

LIST OF DATA ELEMENTS

CHOICE	SCREEN	UPD
-----	-----	---
LCEaaaaaa	List of Elements by Code (starting with Data Element 'aaaaaa').	NO
LNEdaaaaaaaaaaaa	List of Data Elements sorted by name (starting with name 'aaaaaaaaaaaa'). 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
LAEaaaaaaaaaaaa	List of Data Elements sorted by Cobol name (starting with name 'aaaaaaaaaaaa').	NO

LREaaaaaaaaaaaaaaaa	List of Data Elements sorted by relational name (starting with 'aaaaaaaaaaaaaaaa').	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
EaaaaaaXObbbbbbCPccccc	X-references of Data Element 'aaaaaa' to Call of P.M.S.(-CP) of Screen 'bbbbbb' (starting with Macro-Structure 'ccccc').	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
EaaaaaaXPbbbbbbCPccccc	X-references of Data Element 'aaaaaa' to Call of P.M.S. (-CP) of Program 'bbbbbb' (starting with Macro-Structure 'ccccc').	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

EaaaaaaXPbbbbbb9ccccc	X-references of Data Element to Pure COBOL Source Code (-9) of Program 'bbbbbb' (starting with -9 line 'ccccc').	NO
EaaaaaaXFbbbbbb	X-references of Data Element 'aaaaaa' to User Entities (starting with UE 'bbbbbb').	NO

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

All notations between parentheses are optional.

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583	
LIST OF ELEMENTS BY CODE			
ELEM.	PARENT TYPE NAME	INTERNAL FORM.	US LIBR.
ACCTNO	R ACCOUNT NUMBER	9(8)	3 0059
ACTION	R TRANSACTION CODE	X	D 0059
APPLI	R PACBASE APPLICATION CODE	X(03)	D *CEN
CHOICE	R OPERATION CODE	X	D 0059
CITY	R CITY	X(15)	D 0059
CLELE	R PACBASE ERROR MESSAGE KEY	X(17)	D *CEN
DATE	R STANDARD DATE	X(6)	D 0059
ORADDT	R ORDER ACTUAL DELIVERY DATE	X(6)	D 0059
ORDATE	R ORDER DATE	X(6)	D 0059
ORPDDT	R ORDER PROMISED DELIVERY DATE	X(6)	D 0059
ERMSG	R PACBASE ERROR MESSAGE LABEL	X(66)	D 0093
ERUT	R ERUT VECTOR	X	D *CEN
GRAER	R PACBASE ERROR MESSAGE GRAVITY	X	D *CEN
ITPRIC	R ITEM TOTAL PRICE	9(9)V99	3 0059
ITQORD	R ITEM QUANTITY ORDERED	9(6)	3 0059
ITQREC	R ITEM QUANTITY RECEIVED	9(6)	3 0059
LIERR	R PACBASE ERROR MESSAGE LABEL	X(66)	D *CEN
LIGNE	R PRINT LINE FOR BATCH REPORTS	X(132)	D *CEN

O: C1 CH: LCE

		PURCHASING MANAGEMENT SYSTEM	SG000008.LILI.CIV.1583	
LIST OF ELEMENTS BY NAME				
ELEM.	PARENT	T NAME	INTERNAL FORM.	US LIBR.
XKEYXY		R key xy file	X(11)	D *CEN
XZ12		R standard data element 12 char.	X(12)	D *CEN
XZ2		R standard field 2 characters	XX	D *CEN
XZ2A	XZ2	R standard field 2 characters	XX	D *CEN
XZ2B	XZ2	R standard field 2 characters	XX	D *CEN
XZ2C	XZ2	R standard field 2 characters	XX	D *CEN
XZ2D	XZ2	R standard field 2 characters	XX	D *CEN
XZ2E	XZ2	R standard field 2 characters	XX	D *CEN
XZ3		R standard field 3 characters	XXX	D *CEN
XOPSW		R xo password	X(5)	D *CEN
XOAGAC		R Agency address city	X(15)	D *CEN
XOAGAS		R Agency address state	XX	D *CEN
XOAGAZ		R Agency address zip code	X(5)	D *CEN
XOAGA1		R Agency address 1st line	X(20)	D *CEN
XOAGA2		R Agency address 2nd line	X(20)	D *CEN
XOAGC		R Agency code	X(5)	D *CEN
XOAGNM		R Agency name	X(20)	D *CEN
XOAGPA		R Agency phone area code	XXX	D *CEN

O: C1 CH: LNE

LIST ELEMENTS	PURCHASING MANAGEMENT SYSTEM UNDEFINED IN DICTIONARY	SG000008.LILI.CIV.1583
ELEM.		
\$100		
\$2		
\$200		
\$210		
\$30		
ALC		
ALCM		
ALTPCB		
APC		
BB00		
BB10		
BB20		
BLC		
BOTTOM		
BPC		
CCOD		
CF		
CF\$4		
0: C1 CH: LFE		

PURCHASING MANAGEMENT SYSTEM SG000008.LILI.CIV.1583
 ELEMENT X-REFERENCES TO ON-LINE SCREENS FOR ELEMENT : NIVUPA

```

--- SCREEN PA0000 ----- LIBR.
P 25CC520 M XM70-NIVUPA CS00-NIVUTI (1) 0522
P 25FK140 ERB CODUTI XM70-NIVUPA NOT = '0' 0522
--- SCREEN PA0150 ----- LIBR.
CE 230 TYPE: V PRESENTATION 0522
P 07BM160 (V-0150-NIVUPA NOT = 0522
P 07BM170 T-0150-NIVUPA) 0522
P 30BB970 M '0' XM70-NIVUPA XM70-NIVUPA = ' ' 0522
P 65AL520 M ' ' O-0150-NIVUPA O-0150-NIVUPA = '0' 0522
  
```

O: C1 CH: EnivupaX0

PURCHASING MANAGEMENT SYSTEM SG000008.LILI.CIV.1583
ELEMENT X-REFERENCES TO SEGMENTS FOR ELEMENT : PAGE

SEGM.	LIN	SEGMENT NAME	GR	K	LIBR.	
KW	00	090			SEGMENT KW	0522
VC	20	200			TABLE OF CONTENTS AND INDEX LINES	0522
WH	00	110			SEGMENT WH00	0522
WH	17	040			SEGMENT WH17	0522
WK	00	090			SEGMENT WK00	0522
WK	17	040			SEGMENT WK17	0522

0: C1 CH: Epage XS

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583
ELEMENT X-REFERENCES TO REPORTS FOR ELEMENT : PAGE		
REP.	SC LIN PICTURE	LIBR.
DS C	02 074	1391
DS T	02 074	1322
DS X	01 062 IK = IK ADD 1 TO 7-ET00-PAGE	1333
DS 1	01 075 IK = IK ADD 1 TO 7-ET00-PAGE	1322
DS 2	01 075 IK = IK ADD 1 TO 7-ET00-PAGE	1322
DS 9	01 075	1322
ED 7	01 073	1322
ED 7	10 068	1322
EE E	01 111	1336
EE H	03 073	1336
EE L	06 073	1336
EE M	01 075	1336
EE N	01 111	1336
EE O	01 111	1336
EE R	01 111	1336
EE S	01 068	1336
EF G	01 128	1336
EF H	01 128	1336

O: C1 CH: Epage XR

```

-----
                PURCHASING MANAGEMENT SYSTEM                SG000008.LILI.CIV.1583
ELEMENT X-REFERENCES TO PROGRAMS FOR ELEMENT : SERNU

---- PROGRAM   AMSF01 ----- LIBR.
W BB130 05 WW10-SERNU                                0197
P 30BB230 M WW10-SERNU CD10-SERCO                    0126
---- PROGRAM   COGEN6 ----- LIBR
P 30GG110 M AT10-COCLI SE10-COCLI      WW10-SERNU = SE10-SERNU 0126

0: C1 CH: Esernu XP
-----

```

Data Elements: Generation-Print

Lists and description reports on Data Elements may be obtained by entering certain commands on the Generation and Print Commands (GP) screen.

LISTS

LCE: List of all Data Elements, sequenced by code.

C1 OPTION: Without explicit keywords,

C2 OPTION: With explicit keywords.

LKE: List of all Data Elements, by keywords.

After typing LKE, a selection field (SEL:) enables the user to choose implicit ('L') or explicit ('M') keywords, or both (' '). Keywords are entered on a continuation line.

C1 OPTION: Same as LCE.

LNE: List of all Data Elements, sequenced by name (sorted on the first 20 characters and the element codes).

C1 OPTION: Same as LCE.

LXE: List of defined Data Elements having Description lines, Comments lines or Keywords, but not used.

C1 OPTION: Same as LCE.

LAE: List of 'REVERSE' Data Elements by COBOL names.

C1 OPTION: Same as LCE.

DESCRIPTIONS

DCE: Definition, Description and Comments for the Data Element entered in the ENTITY CODE field. If no code is specified, the information on ALL data elements is printed.

C1 OPTION: Definition, description, general documentation without X-references. (Parent and Child Data Elements), C2 OPTION: With X-references.

DFE: Description, X-references of undefined data elements.

C1 OPTION: (Only).

Chapter 3. Data Structures

Data Structures: Introduction

All sets of data processed by applications are described via the Data Structure entity.

Data Structures are logical groups of segments, which may be :

- . Records of files,
- . Segments of databases,
- . Work areas of programs,
- . Inter-program common areas,
- . Table items...

A data structure is described logically, with no physical characteristics. This description may then be reused for different purposes : for example a given segment may be used in a database and in the programs which access the database.

Information of a physical nature (such as organization, block size, description type, etc.) is added when the data structure is used in programs or database blocks.

(See the 'Database Description' and 'Batch Applications' Manuals).

TRANSACTION FILE

Certain files are designed to undergo validation, and then update master (Principal) files or databases.

They are called Transaction Files.

For such files, validation and update procedures are described on the Segment Definition and Call of Elements (-CE) screens (see the 'Batch Applications' Manual).

TABLES

The Data Structures corresponding to Tables are described in the Pactables reference manual.

GENERAL CHARACTERISTICS

The Data Structure entity includes the following:

- A Definition screen (required), for entry of the general characteristics of the Data Structure (name, nature, keywords, etc.),
- A Comments screen (optional). (See chapter 'Documentary Facilities' in the 'Character-Mode User Interface Guide').

- A 'Generation Elements' screen (optional) where you can enter technical information concerning the Data Structure (according to the type of Data Structure, for example, frequency of backups, etc.). (See chapter 'Common Screens' in this Manual).
- An 'Error Messages - Help' screen (optional). (See chapter 'Common Screens' in this Manual).

RESULTS

Once the Data Structures are defined, you can obtain the following:

- A list of all Data Structures,
- Cross-references, listing all uses of the Data Structures in Programs or on-line Screens.

Data Structures: Definition

A Data Structure is defined by its code, name and type.

The 'type' is documentary (mostly used to provide list of Data Structures sorted by type), except for Pactables.

For the Batch Systems Development function, Programs containing the error messages to be used with the application are named in the COMPLEMENT field. For more information see the 'Batch Applications' Manual, Chapter 'Error Messages'.

ASSOCIATED LINES

Comments (-GC).

These lines allow you to insert additional explanatory text.

Generation Elements (-GG).

These lines allow you to indicate technical information about the Data Structure.

Error Messages - Help (-GE).

These lines allow you to specify on-line help on the Data Structure.

```

PURCHASING MANAGEMENT SYSTEM          SG000008.LILI.CIV.1583

DATA STRUCTURE DEFINITION      1  PR

NAME.....: 2 PRODUCT FILE

COMPLEMENT.....: 3

TYPE.....: 4 Z DATA STRUCTURE

EXPLICIT KEYWORDS: 5

SESSION NUMBER.....: 0059          LIBRARY.....: CIV    LOCK:.....:

O: C1 CH: Dpr                      ACTION:

```

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
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	30	NAME OF DATA STRUCTURE (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').
3	44	COMPLEMENT OF DATA STRUCTURE NAME
		With the Batch Systems Development function only:
		Error messages corresponding to validation of a transaction file are coded in at most two programs. Those two program codes are indicated in this field, as follows: Blank in column 1, 'E' in column 2, then one or two program codes.
		Example: Eerrpg1errpg2
		Note: The 'E' is entered in column 36, in batch mode.
		For more information, refer to the Batch Applications Manual, chapter 'Error Messages', subchapter 'Coding of Error Messages'.
4	1	TYPE OF DATA STRUCTURE
		The TYPE of Data Structure is used for documentary purposes, except the Pactables and Logical View types.

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
	'Z' 'G' 'T' 'M' 'N' 'V'	Data structure (Default value) Table (Historical) used with the Pactables function Table (without Historical account), Pactables function Once a Data Structure has been assigned the type 'G' or 'T', this type cannot be changed. Once a Data Structure has been assigned another type, this type cannot be changed to 'G' nor 'T'. Table with historical account, with century. Logical view (See the 'Business Logic' volume of the Developer's Documentation dedicated to eBusiness applications). this type cannot be changed. Once a Data Structure has been assigned another type, this type cannot be changed to 'V'.
5	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'.

Data Structures: On-Line Access

LIST OF DATA STRUCTURES

CHOICE	SCREEN	UPD
-----	-----	---
LCDaa	List of Data Structures by code (starting with Data Structure 'aa').	NO
LNDaa	List of Data Structures by name (starting with Data Structure 'aa').	NO
LTDtaa	List of Data Structures by type (starting with type 't' and D.S. 'aa').	NO
LPDaaaaaa	List of Data Structures by External Name (starting with external name 'aaaaaa').	NO

DESCRIPTION OF DATA STRUCTURE 'aa'

CHOICE	SCREEN	UPD
-----	-----	---
Daa	Definition of Data Structure 'aa'.	YES
DaaCR	Instances linked to Data Structure 'aa' via User Relations.	YES

DaaGCbbb	Comments for Data Structure 'aa' (starting with line number 'bbb').	YES
DaaGEbbb	Error Messages for Data Structure 'aa' (starting with line number 'bbb').	YES
DaaGGbbb	Generation Elements for Data Structure 'aa' (starting with line number 'bbb').	YES
DaaATbbbbbb	Text assigned to the Data Structure 'aa' (starting with text 'bbbbbb').	NO
DaaX	X-references of Data Structure 'aa'.	NO
DaaXQbbbbbb	X-references of Data Structure 'aa' to instances through User Relations (starting with Relation 'bbbbbb').	NO
DaaXVbbbbbb	X-references of Data Structure 'aa' to Documents (starting with Document 'bbbbbb').	NO
DaaXPbbbbbb	X-references to programs for Data Structure 'aa' (starting with program 'bbbbbb').	NO
DaaXPbbbbbbCPccccc	X-references of Data Structure 'aa' to Call of P.M.S. (-CP) of Program 'bbbbbb' (starting with macro-structure 'ccccc').	NO
DaaXPbbbbbbWccddd	X-references of Data Structure 'aa' to Work Areas (-W) of Program 'bbbbbb' (starting with Work Area 'cc', line number 'ddd').	NO
DaaXObbbbbb	X-references of Data Structure 'aa' to Screens (starting with Screen 'bbbbbb').	NO
DaaXObbbbbbCPccccc	X-references of Data Structure 'aa' to Call of P.M.S. (-CP) of Screen 'bbbbbb' (starting with macro-structure 'ccccc').	NO
DaaXObbbbbbWccddd	X-references of Data Structure 'aa' to Work Areas (-W) of Screen 'bbbbbb' (starting with Work Area 'cc', line number 'ddd').	NO
DaaLSbb	Data structure 'aa' list of Segments (starting with Segment 'bb'). .C1: default value. .C2: only the Segment codes and the transaction code values are displayed.	NO

NOTE: After the first choice of type 'Daa', 'Daa' can be replaced with '-'.
All notations between parentheses are optional.

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583
LIST OF DATA STRUCTURES BY CODE		
DS NAME AND COMPLEMENT	T TYPE	LIBR
C* *** TEMPORARY FILES	Z DATA STRUCTURE	0059
CO ORDER PREPARATION	Z DATA STRUCTURE	0059
E* *** REPORTS	J OUTPUT MEDIUM	0059
EO ORDER REPORTS	J OUTPUT MEDIUM	0059
G* *** PERMANENT FILES	Z DATA STRUCTURE	0059
LE PACBASE ERROR MESSAGES	Z DATA STRUCTURE	*CEN
OI PURCHASE ORDER INFORMATION	Z DATA STRUCTURE	0059
PR PRODUCT FILE	Z DATA STRUCTURE	0059
TT TABLE DESCRIPTION	G TABLES	0093
VE VENDOR FILE	Z DATA STRUCTURE	0059
W* *** WORKING ZONES	W WORKING AREAS	0059
X* *** RESERVED FOR LIBRARY CEN	Z DATA STRUCTURE	*CEN
XE REPORTS	J OUTPUT MEDIUM	*CEN
XO Structure for On-Line guide	Z DATA STRUCTURE	*CEN
XW WORKING MODULES	W WORKING AREAS	*CEN
XY WORKING EXTRACTION PROGRAM	Z DATA STRUCTURE	*CEN
*** END ***		
O: C1 CH: LCD		

PURCHASING MANAGEMENT SYSTEM										SG000008.LILI.CIV.1583
LIST OF DATA STRUCTURES BY								EXTERNAL NAME		
EXTERNAL NAME	PC	LC	OAMOU	B	M	U	UNIT	SELECTION	PROGRAM	LIBR.
DC	CD	CO	SSFOU	0	0	R			DVBSIW	0642
DC	CD	CO	SSFOU	0	0	R			DVBSI2	0736
EA	EA	CO	SSFOU	0	0	D		*00	SYNCRO	0683
ED	AB	XE	SSFOU	0	0	I		BL1	TAT443	0775
ED	AB	XE	SSFOU	0	0	I		BL1	TOT65A	0746
ED	AD	XE	SSFOU	0	0	J		BL1	TAT443	0775
ED	AD	XE	SSFOU	0	0	J		BL1	TOT65A	0746
ED	ED	XE	SSFOU	0	0	I		A-Z	AG	0746
ED	ED	XE	SSFOU	0	0	I		A	ESPAI7	0746
ED	ED	XE	SSFOU	0	0	I		F	ES30ED	0169
ED	ED	XE	SSFOU	0	0	I		A	FVBSIE	0125
ED	ED	XE	SSFOU	0	0	I		BV	FVBSIM	0164
ED	ED	XE	SSFOU	0	0	I		B	FVBSIW	0164
ED	ED	XE	SSFOU	0	0	I		S	SYNCRO	0169
ED	ED	XE	SSFOU	0	0	I		BL1	TAT443	0177
ED	ED	XE	SSFOU	0	0	I		BL1	TOT65A	0173
ED	ED	XE	SSFOU	0	0	I		BL1	ZTPUT1	0113
ED	UU	XE	SSFOU	0	0	I		BL1	TAT443	0177

O: C1 CH: LPD

PURCHASING MANAGEMENT SYSTEM SG000008.LILI.CIV.1583
 DATA STRUCTURE X-REFERENCES TO PROGRAMS FOR D.S. : PO

----		PROGRAM	POJ010	-----							LIBR								
		DP	EXTERN	OARFU	BLOCKT	B	M	U	RE	SE	L	UNIT	C	SELECTION	F	E	R	L	PL
CD		PA	PA	SSFOU	ØR										I			1	*DOC
CD		PC	PC	SSFOU	ØR										I			1	*DOC
CD		PD	PD	SSFOU	ØR										I			1	*DOC
CD		PO	PO	SSFIU	ØR										I			1	*DOC

0: C1 CH: Dpo XP


```

-----
PURCHASING MANAGEMENT SYSTEM          SG000008.LILI.CIV.1583
DATA STRUCTURE LIST OF SEGMENTS        MD GENERATION/PRINTING REQUEST

SEGM NAME                               STR. CODE  CMD456  NBENT  OCCUR.  LIBR.
MD1Z USER IDENTIFICATION                * TABLES
MD2Y GENERAT. AND PRINTING PAC TABLES   Z

*** END ***
O: C1 CH: D md LS
-----

```

Data Structures: Generation-Print

Lists and description reports on Data Structures may be obtained by entering certain commands on the Generation and Print Commands (GP) screen.

LISTS

LCD: List of all data structures, sequenced by code.

C1 OPTION: Without explicit keywords,

C2 OPTION: With explicit keywords.

LKD: List of all data structures, by keywords.

After typing LKD, a selection field (SEL:) enables the user to choose implicit ('L') or explicit ('M') keywords, or both (' '). Keywords are entered on a continuation line.

C1 OPTION: Like LCD.

LPD: List of all data structures, sequenced by external name.

C1 OPTION: Like LCD.

LTD: List of all data structures, sequenced by type.

C1 OPTION: Like LCD.

LED: List of error messages, sequenced by Data Structure or Segment.

C1 OPTION: Like LCD.

DESCRIPTION

DCD: Definition, Description and Comments for the data structure entered in the ENTITY CODE field. If no code is entered, the information on ALL data structures will be printed.

C1 OPTION: Provides definition, description and comments, x-references, and a list of associated reports and segments,

C2 OPTION: With the assigned text.

GENERATION REQUESTS

A data structure may be generated to provide a COBOL description which is stored in a source library and may be copied into a program using the COBOL COPY clause.

This description may be used as many times as needed. For more information, see the 'Generation of Copy Books' in the 'Data Dictionary' Manual.

FLD: Specify the flow control for the generation of data structures. You may specify optional commands (CCF:_ and CCB:_ fields) and parameters (on continuation lines).

GCD: Generation and description of a data structure whose code you enter in the ENTITY field. You can indicate optional commands (CCF:_ and CCB:_ fields).

C1 option: generation and description of the data structure without assigned text,

C2 option: generation and description of the data structure with assigned text,

C3 option: besides the generation and description of the data structure, generation of the Data Element and Segment description lines in the form of comments to be used by VAP Connector (an eBusiness tool).

The 'GCD' command also enables you to valorize the parameters of the DATA and/or DATASQ P.I.A. on continuation lines:

- in C4 option, the calls to the DATA and DATASQ P.I.As. will be ignored ;
- in C1, C2 or C3 option, the values entered here will override the values in all the calls to the DATA and DATASQ P.I.As.

Chapter 4. Segments

Segments: Introduction

A Segment entity is made of a structured collection of data elements (elementary or group).

It must belong to a Data Structure.

Each segment, described only once, can be called into any entity that uses segments (programs, screens, segments or database blocks).

A lot of information may be coded on the Segment definition and description lines. This information may be used according to the way the segment is used by other modules of the System (Pactables, BATCH, DBD..).

For information concerning the use of a Segment as a logical view, refer to the 'Business Logic' volume of the Developer's Documentation dedicated to eBusiness applications.

GENERAL CHARACTERISTICS

The Segment entity includes the following:

- A Definition screen (required) for entry of the general characteristics of a segment. Other information may be added if the segment is to be used in batch validation programs.
- A Description screen, to specify the data elements which make up the segment. It is also possible to add validation and updating criteria to be used in batch programs or data useful in the description of database segments, or table items. This information will be added according to the future use of the Segment.
- Comments (optional). (See the 'Character-Mode User Interface Guide', chapter 'Documentary Facilities').
- Generation Options (optional). (See chapter 'Common Screens' in this Manual).
- Generation Elements (optional). (See chapter 'Common Screens' in this Manual).
- Error Messages (optional). (See chapter 'Common Screens' in this Manual).

RESULTS

- Cross-references indicating all the uses of the segment (in Screens, Programs, Segments, Database Blocks),
- Activity calculation, in order to optimize the logical model developed from the conceptual model, when using PACMODEL.

Segments: Definition

SEGMENT DEFINITION

A Segment is defined by its code and name.

The Segment code is made of the Data Structure code and a number.

Depending on future needs, it is also possible to specify:

- the number of occurrences of the Segment (used in the activity calculation of the PACMODEL function),
- the maximum number of items of the table, if the Segment describes a table item.

STANDARD FILES

A standard file may have several types of records.

Nevertheless, the sort criteria and keys must be on all the records. This 'common part' is described once in the Segment number '00'.

The specific part of each record is described in a Segment number 'nn'.

In generated programs, a record description will be made of the concatenation of the '00' and the appropriate 'nn' segment descriptions.

A data element used to identify the specific record type has to be defined on the common part : the CODE OF RECORD TYPE.

This data element code is specified on the definition line of segment number '00'; the appropriate value is coded on the definition line of the specific part segment.

For a file that has only one type of record, a unique '00' segment is described.

TRANSACTION FILE (BATCH SYSTEMS DEVELOPMENT FUNCTION)

A transaction file is made of records that update a 'permanent' file.

A data element belonging to the common part of the file is used to identify the type of update being done (Creation, Modification, Deletion, or other cases). It is called the ACTION CODE.

This Data Element code and values are indicated on the Definition line of the '00' Segment, respectively in the 'CODE OF ACTION CODE' and 'VALUES OF TRANSACTION CODE' fields.

When each specific part Segment is defined, the rules concerning its presence or absence with each type of update are specified in the corresponding fields.

PREREQUISITE

The data structure must have been previously defined.

ASSOCIATED LINES

- Comments (-GC).
These lines are used for documentation purposes.
- 'Generation Elements' (-GG).
These lines are used to customize SQL accesses.
- 'Error Messages - Help' (-GE) where you specify error messages and on-line help on the Segment.
- 'Generation Options' (-GO) for the uppercase-lowercase management in customized SQL accesses.

```

-----
PURCHASING MANAGEMENT SYSTEM          SG000008.LILI.CIV.1583

          1 2
SEGMENT DEFINITION.....: PR00

NAME.....: 3 COMPLETE PRODUCT RECORD

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

CODE OF RECORD TYPE ELEM...: 6
CODE OF ACTION CODE ELEM...: 7
VALUES OF TRANSACTION CODE: CR: 8    MO: 9    DE: 10
                          M4: 11   M5: 12   M6: 13

EXPLICIT KEYWORDS...: 14

SESSION NUMBER.....: 0059      LIBRARY.....: CIV    LOCK.....:

O: C1 CH: Spr00                      ACTION:
-----

```

Segments: On-Line Definition

NUMLEN	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.
		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)

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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.
		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 elm. - table id
		For the Batch Systems Development function:
		CODE OF RECORD TYPE ELEM for the '00' segment:
		Enter the code of the data element used to identify the type of record (left-justified, six characters maximum).
		VALUE OF RECORD TYPE ELEM for the non-00 segments:
		Enter the value to differentiate the individual segments from one another.
		This information is required every time a variable1 file is used in a Segment.
		DL/1, SQL:

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Enter the external name of the segment or object 1 to 8 characters, between quotes).
		For Pactables table segments:
		Enter the END USER TABLE ID on 6 characters.
7	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'.
8	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"
		'F'
		Optional (default).
9	5	MODIFY : 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 "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"

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	'I'	Invalid: the segment must not be present on a "modify"
	'F'	Optional (default)
10	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).
11	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.
	'I'	Invalid: the segment must not be present.
	'F'	Optional (default).
12	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).
13	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:

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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)
14	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'.

Segments: Description

A Segment is described by listing (calling) the Data Elements it contains. This is done by the -CE screen.

Additional information may be coded, according to the future use of the Segment (validation and update for transaction files, keys for database Segments, Pactables information..).

It is highly recommended to dedicate a Segment to only one type of future use.

OPERATION CODE

- C1: default value (Update).
- C2: display of the internal format of the Data Elements.
display of Elements of a called "data aggregate"
(see below).
display of names of Elements defined at the
Segment level.
- C3: display of the input format of each Data Element
called in the Segment.

GENERAL CHARACTERISTICS

A Segment is described by an ordered sequence of Data Elements. This sequence may include group Data Elements, or repetitions of elementary or group Data Elements.

Redefinitions are possible within a Segment.

For files and databases, access and control break sort keys are indicated. Initial values can be defined for work areas.

A segment is described by Data Elements defined in the Specifications Dictionary. As a result, the clear name of the Data Element, its formats and USAGE clauses are channeled down to the Segment level.

It is not possible to modify those characteristics at the Segment level.

It is possible to use Data Element codes which are not defined in the Specifications Dictionary, only when they do not have a real functional meaning (group Elements, fillers, error tables, etc.) In this case, a name and/or a format are required.

It is also possible to describe a Segment containing different aggregates of previously defined data, such as Segments or entities described with the PACMODEL function (Objects and Relationships).

It is not possible to modify the description of the called entity at the Segment level.

The same Data Element code, used in more than one place in a Segment, will provoke generation of identical data names.

PREREQUISITE

The Segment and the Data Elements (except some technical Data Elements which can be defined in the Segment description lines) must have been previously defined.

ASSOCIATED SCREENS

There are additional screens associated (via the LINE NUMBER) with each of the entities called onto the Segment Call of Elements (-CE) screen:

- the S...CEnnnGC screen for comments on the line,
- the S...CEnnnGG screen for additional information about the generation of Database Blocks,
- the S...CEnnnGE screen for additional documentation concerning error messages (Batch Systems Development function).

GROUP ELEMENTS

A Group Element is identified in the list by the number of elementary Data Elements it contains. These Elements are listed after the group element.

A group may include other groups. All elementary Elements are then counted to define the group.

If a dictionary Data Element is used as a group, its length is recalculated (sum of the lengths of the elementary data elements), regardless of its dictionary format.

REDEFINITION

Redefinition is possible within a Segment (generating the COBOL 'REDEFINES' clause). The following is entered in the UPDATE TARGET field:

- . 'R*' in the UPDATE TARGET / FIRST PART,
- . Blank in the rest of the UPDATE TARGET field.

The Data Element containing this option redefines the Data Element of the same COBOL level which precedes it in the Segment description. (See UPDATE TARGET / FIRST PART.)

If a Data Element which redefines another Data Element is contained in a group, it is considered to be an elementary Data Element. It must be taken into account in the calculation of the number of Data Elements contained in a group (except for DL1 database Segments).

NOTE: When Data Elements are redefined, the system does not take their respective lengths into account. This is the user's responsibility.

In the calculation of address length (Segment Level, Address and Length Description (-LAL)), the redefined Data Element length is used for the address calculation.

DATA AGGREGATES

Segments, Model Objects and Relationships (PACMODEL) are also called "data aggregates". They may be called into other segments.

The data aggregate code is indicated instead of the data element code in the list, and it is specified as a special group (see NO. OF ELEMENTARY ELEMENTS IN A GROUP). It may be occurred (See OCCURRENCES (COBOL 'Occurs' clause)).

The description (list of elements) will be included, but it cannot be modified at this level.

NOTE: On the -CE screen, the list of Data Elements of a called aggregate is only viewed in O: C2. When a Segment description is printed (DCS), only the SEGMENT CODE will appear. The expanded view of the Segment may be seen on the Segment Level, Address and Length (-LAL) screen.

LIMITATION

Called Segments may also contain segments. This 'nesting' may occur up to three times.

EXAMPLE:

	ELEM.	GR	01 level: Segment BL00
	ELEM.		01 level: Segment BL00
S BL00 CE	DELCO1		05 level: Delco1
	CL10	**	Segment CL10
S CL10 CE	DELCO2		10 level: Delco2
	DL20	**	Segment DL20
S DL20 CE	DELCO3		15 level: Delco3
	DELCO4		Delco4

	ELEM.	GR	01 level: Segment BL00
			Segment AA30
S AA30 CE	DELCO5	**	20 level: Delco5

DATABASES SEGMENT DESCRIPTION

- Existing DL/1 segments
DL/1 Segments defined prior to the installation of the System may have used Data Element codes that are eight characters in length. This does not conform to the System standards.
In that case, it is possible to define the Elements in the Dictionary to ensure future management in the System, and associate them with the old codes, to maintain compatibility with the existing applications.
- SQL external names
SQL Data element codes are used also by the end-user, so they must be significant. In some cases, a Data Element must be given a code other than its System code.

In these cases, the two codes can be managed as follows:

On the Segment Call of Elements (-CE) screen, enter:

- The data element code in the DATA ELEMENT CODE field,
- 'A*' in the UPDATE TARGET / FIRST PART field,
- The former code (up to 8 characters) in the UPDATE TARGET / SECOND and LAST PARTs.

For DL/1, the 'old' code will be not only used in the Database Block description, but also in generated SSAs for on-line or batch programs.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	'D'	Deletion of the line
	'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 processing
4	3	Line number
		Numeric.
		It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
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
		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.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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).
		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.
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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		It is required only in the following cases :
		- For an elementary Data Element not defined in the Dictionary (COBOL format),
		- 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 or sort key
		This field identifies all data elements that might be used as control break sort keys, or as access keys to a file, a database or a Pactables table.
		Note: It is highly recommended to dedicate a Segment to only one type of use.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Each data element that may belong to a sort key must be referenced by a unique alphabetic or numeric character. It is recommended to reference the indicators by a series (1, 2, 3 ...).
		The actual sort sequence will be chosen at the program level (on the Call of Data Structures (-CD) screen) by sequencing the characters in the appropriate order.
		Reminder:
		The format of key group data elements must have been entered in the Dictionary or at the segment level.
		PACTABLES:
	'U'	References the access key for a VisualAge Pacbase table. This value must be indicated on the group data element if it is a group key.
	'S'	Indicates that the data element belongs to at least one sub-system.
		DL1 DBD (See the DL/1 DATABASE DESCRIPTION Reference Manual)
	'U'	References a unique key for an DL/1 database.
	'M'	References a multiple key for an DL/1 database.
	1 to 9	Secondary index
		All other values designate a search field.
		DBD AS400 physical file (See the corresponding DBD Reference Manual)
	0 to 9	AS400 physical file key.
		Relational databases (See the corresponding DBD Reference Manual)
	'V'	Variable length column
	'Blank'	Fixed length column
	'W'	For DB2 SQL, SQL/DS and ORACLE, generation of a variable length column (VARCHAR).
	'L'	For DB2 SQL, SQL/DS and ORACLE, generation of a LONG VARCHAR.
		NOTE: Sort keys are not allowed on data elements redefining other data elements (see VALIDATION and UPDATE FIELDS, below).
12	30	VALIDATION AND UPDATE FIELDS
		This field is made of four main fields :
		- 'CMD456' reserved to presence validation - 'CONT' reserved to class and value validation - 'VALUE/FCT' complementary to the previous one - 'UPD/TRGET' reserved to batch update
		Those fields are mostly used in batch validation and update programs; their detailed description is to be found in the 'Batch Applications' Manual.
		Nevertheless, they may be used for other purposes:
		DEFINITION OF AN INITIAL VALUE
		.'T' column of the 'CONT' field :

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			- 'V' definition of an initial value, which is specified as a literal in the 'VALUE/FCT' field or on the element description (type 'D' line) (10 characters).
			- 'W' same as 'V', but the literal may continue in the 'UPD/TRGT' field (10 more characters).
			. The initial value will be taken into account in programs and also in the generation of COPY BOOKS (COBOL value clause), if needed.
			REDEFINITION
			. 'UPD/TRG' field : enter R*, left justified. The remaining part of the field should be blank. It means the data element redefines the preceding data element of the same level.
			Pactables function:
			In the 'T' column of the 'CONT' field :
		'S'	Indicates that the element belongs to one or more sub-schemas. Sub-schemas are indicated by entering the letter 'O' positionally in the VALUES/FCT field as illustrated below:
			Example:
			ELEM. CONT VALUE/SFC
			delco S O OOO
			The data element 'delco' belongs to sub-schemas 1,3,4 and 5.
			SQL RELATIONAL DBD: 'C' column of the 'CMS456' field :
			Indicates the presence of a Table Column.
			.In the 'T' column of the CONT field :
		'S'	Indicates that the Data Element belongs to one or more sub-schema(s).
			.VALUES/FCT field :
			Indicates to which sub-schema(s) the Data Element belongs.
			.UPD/TRG field :
			The relational name of a Column may be entered in this field.
			For more information, refer to Subchapter "Table or View Description" in the "Relational / SQL Database Description" manual.
13	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).

Segments: On-Line Access

LIST OF SEGMENTS

CHOICE	SCREEN	UPD
LCSaaaa	List of Segments by code (starting with Segment 'aaaa').	NO
LNSaaaa	List of Segments by name (starting with Segment 'aaaa').	NO

DESCRIPTION OF SEGMENT 'aaaa'

CHOICE	SCREEN	UPD
Saaaa	Definition of Segment 'aaaa'.	YES
SaaaaCR	Instances linked to Segment 'aaaa' via User Relations.	YES
SaaaaGCbbb	Comments on Segment 'aaaa' (starting with line number 'bbb').	YES
SaaaaGEbbb	Error messages on Segment 'aaaa' (starting with line number 'bbb').	YES
SaaaaGGbbb	Generation Elements for Segment 'aaaa'(starting with line number 'bbb').	YES
SaaaaGObbb	Generation option for Segment 'aaaa' (starting with line number 'bbb').	YES
SaaaaATbbbbbb	Text assigned to Segment 'aaaa' (starting with text 'bbbbbb').	NO
SaaaaLSPbbb	List of Parent Segments for Segment 'aaaa' (starting with Parent Segment 'bbbb').	NO
SaaaaLSCbbb	List of Child Segments for Segment 'aaaa' (starting with Child Segment 'bbbb').	NO
SaaaaX	X-references of Segment 'aaaa'.	NO
SaaaaXSbbb	X-references of Segment 'aaaa' to segments (starting with Segment 'bbbb').	NO
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
SaaaaXPbbbbbbCPccccc	X-references of Segment 'aaaa' to Call of P.M.S. (-CP) of Program 'bbbbbb' starting with Macro-Structure 'ccccc').	NO
SaaaaXPbbbbbbWccddd	X-references of Segment 'aaaa' to Work Areas (-W) of Program 'bbbbbb' (starting with Work Area 'cc', line number 'ddd').	NO
SaaaaXObbbbb	X-references of Segment 'aaaa' to Screens (starting with Screen 'bbbbbb').	NO

SaaaaX0bbbbbbCPccccc	X-references of Segment 'aaaa' to Call of P.M.S.(-CP) of Screen 'bbbbbb' (starting with Macro-Structure 'ccccc').	NO
SaaaaX0bbbbbbWccnnn	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.

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583	
LIST OF SEGMENTS BY CODE			
CODE	NAME OF THE SEGMENT OR D.S.	TYPE OF THE D.S.	LIBR.
CO	ORDER PREPARATION	Z DATA STRUCTURE	0059
C000	ORDER ITEM		0059
LE	PACBASE ERROR MESSAGES	Z DATA STRUCTURE	*CEN
LE00	PACBASE ERROR MESSAGES		*CEN
OI	PURCHASE ORDER INFORMATION	Z DATA STRUCTURE	0059
OI00	PURCHASE ORDER KEYS		0059
OI10	BASIC ORDER DATA		0059
OI20	ORDER LINE ITEM DATA		0059
PR	PRODUCT FILE	Z DATA STRUCTURE	0059
PR00	COMPLETE PRODUCT RECORD		0059
PR01	PRODUCT INFORMATION		0059
TT	TABLE DESCRIPTION	G TABLES	0093
TT20	AREA CODES		0093
VE	VENDOR FILE	Z DATA STRUCTURE	0059
VE00	VENDOR INFORMATION		0059
X0	STRUCTURE FOR ON-LINE GUIDE	Z DATA STRUCTURE	*CEN
X001	PASSWORD		*CEN
X002	ROOT SEGMENT		*CEN

O: C1 CH: LCS

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583				
SEGMENT X-REF'S TO BLOCKS FOR SEGMENT : MV00						
BLOCK	NAME	LIN	TY	AR/SET	PRNT CHLD	OCC. LIBR.
DM4M1	TEST	200	M3	AREA01		0 0431
DM4M1	TEST	300	M3	SET001	MV00 MV01	0 0431

0: C1 CH: -XB

PURCHASING MANAGEMENT SYSTEM SG000008.LILI.CIV.1583
 DESCRIPTION OF SEGMENT : PR00 COMPLETE PRODUCT RECORD

A	LIN	LEVEL	ELEM.	OCC	INT.	FOR.	U	LGTH	ADD	INP.	FOR.	LGTH	ADD
	000	10	PRDKEY						1				
	010	11	VENUMB		X(5)		D	5	1	X(5)		5	
	020	10	PR01										
	100	11	PRNUMB		X(10)		D	10	6	X(10)		10	
	110	11	PRDESC		X(30)		D	30	16	X(30)		30	1
	120	11	PRPRIC		9(6)V99		3	5	46	9(6)V99		8	4
	130	11	PRDTIM		999		3	2	51	999		3	5
	140	11	PRMEAS		XX		D	2	53	XX		2	5

*** END ***
 0: C1 CH: Spr00LAL

PURCHASING MANAGEMENT SYSTEM SG000008.LILI.CIV.1583
 DESCRIPTION OF SEGMENT : PR00 COMPLETE PRODUCT RECORD

A ELEM.	NAME	INP. FOR.	INT. FOR.	U	OCC	GR	K	LIBR.
	PRDKEY PRODUCT KEY						1	U 0059
	VENUMB VENDOR NUMBER	X(5)	X(5)		D		B	0059
	PR01						**	0059
	PRNUMB PRODUCT NUMBER	X(10)	X(10)		D		A	0059
	PRDESC PRODUCT DESCRIPTION	X(30)	X(30)		D			0059
	PRPRIC PRODUCT PRICE	9(6)V99	9(6)V99		3			0059
	PRDTIM ESTIMATED DELIVERY TIME	999	999		3			0059
	PRMEAS UNIT OF MEASURE	XX	XX		D			0059

*** END ***
 0: C1 CH: -DED

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583
DESCRIPTION OF SEGMENT : PR00 COMPLETE PRODUCT RECORD		
	PR00	TOTAL
NUMBER OF DATA ELEMENTS.....:	8	8
NUMBER OF ELEMENTARY FIELDS...:	6	6
INPUT LENGTH.....:	58	58
INTERNAL LENGTH.....:	54	54
*** END ***		
0: C1 CH: -STA		

Segments: Generation-Print

Lists and description reports on data structures may be obtained by entering certain commands on the Generation and Print Commands (GP) screen.

LISTS

LCS: List of Segments sequenced by code.

C1 OPTION: Without explicit keywords,

C2 OPTION: With explicit keywords.

LKS: List of Segments sequenced by keyword.

After typing LKS, a selection field (SEL:) enables the user to choose implicit ('L') or explicit ('M') keywords, or both (' '). Keywords are entered on a continuation line or

DESCRIPTION

DCS : Segment description On the GP screen, enter the Data Structure code in the ENTITY field. The segment selection is made by listing the 2-characters numbers (00,10,20..) on the continuation line. To get the continuation line, put an '*' in the 'S' field.

The format of the Elements may be selected. After typing 'DCS', a FORMAT: field appears.

The valid values are :

.I = internal,

.E = input,

.S = output.

.R = internal, but if there is a relational

format, it replaces the Data Element format.

Regardless of the selected Library code, the print option for this entity can only be '1' or '2' (C1, U1, etc., C2, U2, etc.).

Option '1' generates the printing of:

- The definition line of the data structure:
Associated keywords and general comments lines,
Cross-references to programs and screens,
The list of segments belonging to the data structure,
- The definition line of each segment:
Associated keywords and comments lines,
Cross-references to all other entities,
- Description lines of each segment:
The list of sub-schemas and sub-systems (Pactables only)
The call of elements (including the comments),
The statistics of the segment (number of elementary elements and record length).

NOTE: For table segments, see the Pactables Reference Manual.

Option '2' provides the same listings as above, but adds a listing of the texts assigned to the data structure and the segment.

Chapter 5. Database Blocks

Database Blocks: Introduction

The purpose of the Database Block entity is to:

- Describe, at the logical level, hierarchical, relational or network databases,
- Describe, at the physical level, hierarchical, relational, or network databases, taking into account the characteristics of certain DBMSs.

GENERAL CHARACTERISTICS

The definition and description of logical or physical segments is ensured by the Segment entity.

The definition and description of logical or physical relationships between segments is ensured by the Database Block entity.

The Database Block entity includes the following:

- A Definition (required), for entry of the general characteristics (clear name, type, external name, keywords, etc.),
- Descriptions, which vary according to the type of block,
- Comments (optional). (See the 'Character-Mode User Interface Guide', chapter 'Documentary Facilities').
- Generation options (optional). (See chapter 'Common Screens' in this Manual).
- Generation elements (optional). (See chapter 'Common Screens' in this Manual).

NOTE

The actual generation of these descriptions is not accomplished by the Specifications Dictionary function.

The Database Description function actually generates these descriptions according to the syntax required by the DBMS.

The Database Block may be linked to a data model built with the PACMODEL function.

Database Blocks: Definition

A database block is defined by a code, a name and a type.

The Definition is done on a 'B' screen.

There are three categories of Database Blocks:

- Hierarchical blocks used to describe tree-like structured hierarchical databases (for example, DL/1).
- Network blocks used to describe the relationships in network databases (for example, CODASYL).

This category also allows for the description of TANDEM or DB2 databases.

- Relational blocks used to describe the links in relational databases in the SQL language (for example, DB2).

To each one of these block categories corresponds a specific Description.

A Database Block is classified into one of these categories according to the TYPE OF BLOCK. A change in the block type is allowed only when it does not imply a change in the block's category.

ASSOCIATED LINES

Comments lines.

The definition and description lines of a Database Block support all of the logical information necessary for the generation of the block in source language.

The physical level information is entered on the 'Generation Elements' (-GG) screen associated with the definition and description lines of the Database block.

Options are entered on the 'Generation Options' (-GO) screen.

This can be facilitated by the use of pre-defined Parameterized Input Aids (P.I.A.'s).

```

PURCHASING MANAGEMENT SYSTEM          SG000008.LILI.CIV.1583
BLOCK DEFINITION.....: 1 ORDRDB
NAME.....: 2 ORDER DATABASE
TYPE.....: 3 D1 SCHEMA (DDL)
VERSION.....: 4

EXTERNAL NAME.....: 5 ORDRDB00
EXT. NAME OF SCHEMA : 6

CONTROL CARDS..... FRONT: 7 L   BACK: 8 L

EXPLICIT KEYWORDS..: 9 IMS

SESSION NUMBER.....: 0059          LIBRARY.....: CIV   LOCK....:

O: C1 CH: Bdbordb                      ACTION:

```

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
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.
	'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).

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
		RELATIONAL DATABASES
	Q2	DB2 SQL
	Q3	SQL SERVER
	Q4	DB2/400
	QA	ALLBASE/SQL
	QB	DB2/2 and DB2/6000
	QC	DATACOM/DB
	QG	INGRES/SQL
	QI	INFORMIX-ESQL
	QN	NONSTOP SQL
	QO	ORACLE (releases earlier than V6)
	QP	ORACLE (from release V6 on)
	QR	RDMS
	QS	SQL/DS
	QT	INTEREL RDBC
	QU	INTEREL RFM
	QV	VAX SQL
	QY	SYBASE
	DB	DB2 (It is recommended to use the Q2 type)
		NETWORK DATABASES
		.CODASYL-DM4 (BULL 66 or DPS8):
	'M1'	DDL schema, only elementary fields are generated,
	'M4'	DDL schema, only group fields are generated,
	'M2'	DMCL schema,
	'M3'	Sub-schema.
		.CODASYL-IDS2 (BULL 64 or DPS7):
	'I1'	DDL schema,
	'I2'	DMCL schema,
	'I3'	SDDL sub-schema.
		.CODASYL-IDMS:
	'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

NUM	LEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
			AS/400 PHYSICAL FILE
		PF	AS/400 Physical file (IBM SYS. 38)
		LF	AS/400 Logical file (IBM SYS. 38).
			DDL TURBOIMAGE
		TI	TurboImage Database.
			DMSII DATABASE
		20	DMSII Database (DASDL)
4	4		VERSION NUMBER
			Version number of the database system.
		2000	DB2/400 : Version 2
		3000	NONSTOP SQL: Version C30
		4000	VAX SQL : Version 4.0
		5000	RDMS 1100 : Version 5RA4
		7000	ORACLE : V7
		Blank	Other systems, all versions.
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	8		EXTERNAL NAME OF THE SCHEMA
			This field is only used for SE-type blocks (Group of Sets) and for CODASYL Blocks. Otherwise, it is not displayed.
			This is necessary at generation time if the block is a SUB-SCHEMA or a DMCL.
			This is the physical name of the schema to which the given block is attached.
			This field is not used if the block is a schema.
7	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.
8	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.
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).

NUM	LEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
			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'.

Hierarchical Database Blocks: Description

The -DH screen is used to describe the relationships between segments in a hierarchical database (example: DL/1).

GENERAL CHARACTERISTICS

Each line designates a segment and its parent (except for the line referring to the root segment). The exact position of the segment within the hierarchical structure is indicated according to normal DL/1 standards, that is top to bottom, left to right.

PREREQUISITE

The hierarchical block must be defined as well as all the entities called.

ASSOCIATED SCREEN

The 'Comments' screen (-GC) to document each description line.

The 'Generation Elements' screen (-GG) is used to provide the physical information necessary in order to generate the block. This screen is associated with each description line and is accessed by the choice '-DHnnnGG' (where 'nnn' represents the description LINE NUMBER of the entity concerned).

OPERATION FIELD

C1: default value.

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
		With the PACMODEL function only:
		Enter the code of the MODEL ENTITY RELATIONSHIP that defines the link between the segment and its parent.
		The System will automatically create a cross-reference for these relationships.
		NOTE: The relationships are described via the PAC- MODEL function.
7	1	KEY INDICATOR
		Used for a symbolic reference of the key data element of a given segment in a given DBD. The character indicated in this field must also appear on the Segment Call of Elements (-CE) screen in the KEY INDICATOR FOR ACCESS OR SORT field, on the key data element line.
		'U'
		References a unique key.
		'M'
		References a multiple key.
		1 to 9
		DL/1 Secondary index.
		\$
		In a PCB or a physical or logical DBD (Block type PC, DB, or DL), generates a non-qualified SSA (used in OLSD).
		All other values designate a search field.
		NOTE: Sort keys are not permitted on data elements redefining other data elements (see the Segment Call of Elements (-CE)).
8	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).
9	5	EST. NUMBER OF CHILD / PARENT LINK
		PURE NUMERIC FIELD
		This gives the average number of occurrences of child segments that are linked to an occurrence of a parent segment. This number is used for activity calculation (see the PACMODEL Reference Manual).
10	36	COMMENT/RELATIONSHIP/KEYLENGTH
		Documentary purpose mostly : it allows to document the defined parent/child link.
		It is also used to indicate the concatenated key length (cc=...).

Relational/SQL Database Blocks: Description

The -DR screen allows the description of a Database Block's physical model (i.e. table spaces, tables, table views, index or key) used to generate an SQL database description.

On each description line, a special ACTION CODE indicates the type of SQL command to generate for the block's object.

Description lines which are assigned Comments (-GC) or Generation Elements (-GG) are displayed with a '*'.

PREREQUISITE

The Database Block, as well as all called entities, must have been previously defined.

OPERATION FIELD

C1: only value.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	V	View
	I	Index
	A	Alter Table: Column updating
	K	RDMS 1100, Allbase/SQL: Primary Key (Processed with the generation of the table that precedes it.)
		DB2, Datacom/DB, Informix-ESQL, 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, Informix, Sybase and SQL Server: Foreign key (Processed with the generation through an ALTER TABLE command.)
		Allbase/SQL: Foreign Key (Processed with the generation of the table that precedes it.)
	C	Package (Oracle V7 only)
	E	Function (Oracle V7 only)
	Q	Procedure (Oracle V7, Ingres, Sybase, SQL Server and Informix)
	R	Oracle V7, Sybase and SQL Server: Trigger
		Ingres/SQL: Rule
5	27	DATABASE OBJECT EXTERNAL NAME
		It is the name used by the end-user.
		It is prohibited for a Primary Key (K-type line, DB2, DB2/2, DB2/6000 or DATACOM/DB).
		It is required for a Tablespace (P-type line).
		For all other objects, this name may be defined at several levels.
		The priority, at generation time, will be as follows:
		- the external name defined here (-DR),
		- or the one defined in the CODE OF RECORD TYPE ELEMENT field on the Segment Definition screen, defining the corresponding object.
		- or the code of the Segment defining the corresponding object.
		For a Foreign Key (J-type line), two separate codes are required: the constraint name (8 char. maxi) and the Segment code of the reference Table.
6	4	TABLE OR VIEW CODE
		On a T, V or A-type line, this field contains the code of the Segment which represents the Columns of the Table or View.
		On an I, K or J-type line, this field contains the code of the Segment which supports the key.
		On a P-type line, this field must be left blank.
7	1	KEY TYPE
	U	On an I-type line: This value is entered in order to generate the UNIQUE command. Value '0' corresponds to sub-schema '10'.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	0-9	On a V-type line: View of the sub-schema Data Element selection in the Segment. Value 0 corresponds to the sub-schema 10.
	*	All Data Elements of the Segment are included in the View.
	R C S	On a J-type line: Restrict (default value for DB2-type Databases only) Cascade (DB2, SQL/DS, DB2/2, DB2/6000 and ORACLE V7 only) S: Set null (DB2, SQL/DS, DB2/2 and DB2/6000 only)
	B BLANK	On a C-type line (ORACLE V7): Indicates the package type. BODY package standard package
	A B	On a R-type line: Indicates where the trigger or the rule starts to operate: After Before
8	1	TYPE OF GENERATED TRANSACTION
		This field is entered in order to generate the following SQL commands: CREATE, ALTER, and DROP.
	C	CREATE Default value when the corresponding line is created. (No other value may be entered on A-type lines).
	M	ALTER (Except for SQL/400 and INGRES/SQL) Only taken into account for on-line generation through the '-GN' screen. Not allowed on K and J-type lines, except for DATACOM, where a K-type line generates an ALTER TABLE ADD PRIMARY KEY command.
	D	Cancellation: generation of a DROP command. For J (Foreign Key) and K (Primary Key) lines, a DROP PRIMARY KEY or DROP FOREIGN KEY command is generated in an ALTER TABLE command.
	Blank	No generation (-GN); no generation through the GPRT procedure with option 'C2'.

CODASYL, TANDEM And DB2 Blocks: Description

The -DC screen is used to logically describe a CODASYL schema or sub-schema, i.e.:

- declare areas,
- call records and distribute them among areas,
- define and describe sets (code, name, owner record, member record).

By default, a record is mono-area. Should a record be described as multi-area, its description must be overridden by a Comment line (-DCnnnGC, where nnn is the line number).

DESCRIPTION OF A DB2 OR TANDEM DATABASE BLOCK

PRELIMINARY NOTE: The 'Q2'-type Database Block - used to generate the SQL description of relational databases - is to be used.

The 'DB'-type block corresponds to the first version of the DBD DB2 module.

On the Description screen of a DB2 or TANDEM Database Block, the user calls table views, tables or table spaces.

'Q2'-type blocks are described in the previous subchapter 'Description of a relational block.'

PREREQUISITE

The Database Block, as well as all the called entities, must have been previously defined.

OPERATION FIELD

- C1: default value.
- C2: source display.

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
	'A'	Area.
5	6	AREA OR SET CODE (REQUIRED)
		CODASYL:
		In this field, the user enters the code which corresponds to the selected description line type.
		Type 'S': Set code (6 characters), Type 'A': Area code (6 characters), Type 'R': Code of area to which the record belongs.
6	4	OWNER SEGMENT CODE
		With TYPE = 'A': Not used.
		With TYPE = 'R': Enter the code of the segment.
		With TYPE = 'S': Enter the parent segment code (OWNER).
7	4	MEMBER SEGMENT CODE
		With TYPE = 'S', enter the child segment code (MEMBER).
8	6	MODEL RELATIONSHIP CODE
		SCHEMA
		Used only with TYPE = 'S'.
		With the Methodology function only:
		Enter the Relationship code from which the set is derived. VA Pac will automatically create a cross-reference for these relationships.
		NOTE: The relationships are described via the Methodology Function.
		SUB-SCHEMA
		Only used for IDMS ('D3', 'D4' types), DM4 ('M3' type) and DMS ('S3' type) sub-schemas.
		for 'R'-type lines :
		It is possible to change the description of the selected record. The user must indicate the code of the segment redefining the selected segment, as follows : '=FFnn'.
		EXAMPLE:
		T AREA OWNER MEM METHOD OCC NAME OF AREA, SET SEG SEG CODE SET OR COMMENT R AREA1 FF10 =FF20
		In this example record FF10 is generated with the elements belonging to FF20.
		NOTE: Segment FF20 must have been previously defined and described.
9	5	NUMER.
		NUMBER OF OCCURRENCES OF SETS
		PURE NUMERIC FIELD
		Used only with TYPE = 'S':
		This is the average number of occurrences of MEMBER segments that are linked to an occurrence of an OWNER segment. This number is used for Activity Calculation (see the PACMODEL Reference Manual).
10	36	
		NAME OF AREA, SET, OR COMMENT

NUM	LEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
			With TYPE = 'S': Set name, With TYPE = 'A': Area name, With TYPE = 'R': Comment.
			SUB-SCHEMA IDMS (D4) OR DMS (S3):
			There are four different ways to select a record sub- set, as illustrated in the following example:
			LIN : T AREA OWNER MEM MODEL OCC NAME OF AREA, SET SEG SEG CODE SET OR COMMENT 001 : R AREA1 FF10 002 : R AREA1 FF10 =FF20 003 : R AREA1 FF10 SS=n 004 : R AREA1 FF10 =FF20 SS=n
			LINE 001: Record FF10 of the sub-schema is made up of all the data elements of Segment FF10.
			LINE 002: Record FF10 of the sub-schema is made up of all the data elements of Segment FF20.
			LINE 003: Record FF10 of the sub-schema is made up of the data elements of Sub-schema n.
			LINE 004: Record FF10 of the sub-schema is made up of the data elements of Sub-schema n of Segment FF20.
			IDS2 (I3) sub-schema:
			It is possible to call an object (area, record, set) without re-describing it, by specifying: INCLUSION.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		When the Database Block is generated, this code appears after the 'RECORD' clause.
6	36	FILE NAME (REQUIRED)
		This field contains the name of the physical file which supports the data.
		When the Database Block is generated, this name appears after the 'FILE IS' clause.

Database Blocks: On-Line Access

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').	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 'aaaaaa' (starting with text 'bbbbbb').	NO
BaaaaaaX	Cross-references of Database Block 'aaaaaa'.	NO
BaaaaaaXBbbbbbb	Cross-references of Database Block 'aaaaaa' to PSB's (starting with PSB 'bbbbbb').	NO
BaaaaaaXObbbbbb	Cross-references of Database Block 'aaaaaa' to Screens (starting with Screen 'bbbbbb').	NO
BaaaaaaXObbbbbbCSdddd	Cross-references of Database Block '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.	NO

BaaaaaaXObbbbbbWccddd	Cross-references of Database Block 'aaaaaa' to the Work Areas of Screen 'bbbbbb' (starting with Work Area 'cc', line number 'ddd').	NO
BaaaaaaXQbbbbbb	List of occurrences linked to Database Block 'aaaaaa' through User-Defined Relation (starting with Relation 'bbbbbb').	NO
BaaaaaaXVvvvvvv	Cross-references of Database Block 'aaaaaa' to Volumes (starting with Volume 'vvvvvv').	NO
BaaaaaaXPbbbbbb	Cross-references of Database Block 'aaaaaa' to Programs (starting with Program 'bbbbbb').	NO
BaaaaaaXPbbbbbbWccddd	Cross-references of Database Block 'aaaaaa' to Work Areas of Program 'bbbbbb' (starting with Work Area 'cc', line number 'ddd').	NO
CODASYL (NETWORK) DATABASE BLOCK DESCRIPTION		
CHOICE	SCREEN	UPD
-----	-----	---
BaaaaaaDCbbb	Description of CODASYL Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaDCbbbGCccc	Comments on CODASYL Database Block 'aaaaaa' description line 'bbb' (starting with Comments line 'ccc').	YES
BaaaaaaDCbbbGGccc	Generation Elements on CODASYL Block 'aaaaaa' description line 'bbb' (starting with line 'ccc').	YES
LCAaaaaaa	List of areas by code (starting with area 'aaaaaa').	NO
LCCaaaaaa	List of CODASYL sets (starting with set 'aaaaaa').	NO
CaaaaaaACT	CODASYL activity on a set (starting with set 'aaaaaa').	NO
BaaaaaaDCbbb	Description of DB2 Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaDCbbbGCccc	Comments on DB2 Database Block 'aaaaaa' description line 'bbb' (starting with Comments line 'ccc').	YES
BaaaaaaDCbbbGGccc	Generation Elements for DB2 Database Block 'aaaaaa' description line 'bbb' (starting with line 'ccc').	YES
BaaaaaaSQL	Interactive SQL for consultation and update of DB2 catalog for Database block 'aaaaaa'.	NO
TANDEM DATABASE BLOCK DESCRIPTION		
CHOICE	SCREEN	UPD
-----	-----	---
BaaaaaaDCbbb	Description of TANDEM Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaDCbbbGCccc	Comments of TANDEM Database Block 'aaaaaa' description line 'bbb' (starting with line 'ccc').	YES
BaaaaaaDCbbbGGccc	Generation Elements for TANDEM Block 'aaaaaa' description line 'bbb' (starting with line 'ccc').	YES

BaaaaaaDHbbb	Description of Hierarchical Database Block 'aaaaaa' (starting with line 'bbb')	YES
BaaaaaaDHbbbGCccc	Comments on Hierarchical Database Block 'aaaaaa' description line 'bbb' starting with Comments line 'ccc').	YES
BaaaaaaDHbbbGGccc	Generation Elements for Hierarchical Block 'aaaaaa' description line 'bbb' starting with line 'ccc').	YES

LIST OF RELATIONAL/SQL OBJECTS

CHOICE	SCREEN	UPD
-----	-----	---
LSTddss	List of Relational/SQL Objects by type and code (starting with with type 't', code 'ddss').	NO
LESteeeeeeeeeee	List of Relational/SQL Objects by type and external name (starting with type 't' and external name 'eeeeeeeeeee'). Note: If the external name is indicated on the Segment definition, it is not taken into account in the list.	NO

RELATIONAL/SQL DATABASE BLOCK DESCRIPTION

CHOICE	SCREEN	UPD
-----	-----	---
BaaaaaaDRbbb	Description of Relational/SQL Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaDRbbbGCccc	Comments on Relational/SQL Database Block 'aaaaaa' description line 'bbb' (starting with Comments line 'ccc').	YES
BaaaaaaDRbbbGGccc	Generation Elements of Relational/SQL Block 'aaaaaa' description line 'bbb' (starting with line 'ccc').	YES
BaaaaaaDRbbbK	Building of Relational/SQL key 'K' on description line 'bbb' of Block 'aaaaaa'.	YES
BaaaaaaGN	Generation of SQL commands for Relational/SQL block 'aaaaaa'.	YES
BaaaaaaGNnnn	Generation of SQL commands for the Object defined on description line 'nnn' of Block 'aaaaaa'.	YES

NOTE: After the first choice of type 'Baaaaaa', 'Baaaaaa' can be replaced with '-'.
All notations between parentheses are optional.

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583	
LIST OF BLOCKS BY CODE			
CODE	NAME	T TYPE	LIBR.
ALTPCB	ALTERNATE PCB	PC PCB	*CEN
EXPPCB	EXPRESS PCB	PC	*CEN
INDEX1	PRIMARY INDEX	IP PRIMARY INDEX	0093
LEDBD	ERROR MESSAGE DBD	DP PHYSICAL DBD	*CEN
LEPCB	ERROR MESSAGE PCB	PC PCB	*CEN
LEPSB	ERROR MESSAGE PSB	PS PSB	*CEN
ORDRDB	ORDER DATABASE	DP PHYSICAL DBD	0059
ORDRPC	ORDER PCB	PC PCB	0059
ORDRPS	ORDER PSB	PS PSB	0059
PX0010	PCB FOR XO DIALOGUE	PC PCB	*CEN
PX0020	PCB FOR XO DIALOGUE	PC	*CEN
SPCH01	LOGICAL SCHEMA	M1 SCHEMA (DDL)	0093
USPCB	TEST PCB	PC PCB	0093
VEPRDB	VENDOR PRODUCT DATABASE	DP PHYSICAL DBD	0059
VEPRPC	VENDOR PRODUCT PCB	PC PCB	0059
X00010	PSB FOR XO DIALOGUE	PS PSB	*CEN
*** END ***			
0: C1 CH: LCB			

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583		
LIST OF BLOCKS BY TYPE				
T TYPE	CODE	NAME	LIBR.	
DP PHYSICAL DBD	LEDBD	ERROR MESSAGE DBD	*CEN	
	ORDRDB	ORDER DATA BASE	0059	
	VEPRDB	VENDOR PRODUCT DATA BASE	0059	
IP PRIMARY INDEX	INDEX1	PRIMARY INDEX	0093	
M1 SCHEMA (DDL)	SPCH01	LOGICAL SCHEMA	0093	
PC PCB	ALTPCB	ALTERNATE PCB	*CEN	
	EXPPCB	EXPRESS PCB	*CEN	
	LEPCB	ERROR MESSAGE PCB	*CEN	
	ORDRPC	ORDER PCB	0059	
	PX0010	PCB FOR XO DIALOGUE	*CEN	
	PX0020	PCB FOR XO DIALOGUE	*CEN	
	USPCB	TEST PCB	0093	
	VEPRPC	VENDOR PRODUCT PCB	0059	
	PS PSB	LEPSB	ERROR MESSAGE PSB	*CEN
		ORDRPS	ORDER PSB	0059
X00010		PSB FOR XO DIALOGUE	*CEN	
*** END ***				
0: C1 CH: LTB				

PURCHASING MANAGEMENT SYSTEM		SG000008.LILI.CIV.1583	
LIST OF BLOCKS BY EXTERNAL NAME			
TYPE	EXT NAME	NAME	CODE LIBR.
DP PHYSICAL DBD	LEDBD	ERROR MESSAGE DBD	LEDBD *CEN
DP	ORDRDB00	ORDER DATA BASE	ORDRDB 0059
DP	VEPRDB00	VENDOR PRODUCT DATA BASE	VEPRDB 0059
IP PRIMARY INDEX	DBDINDEX	PRIMARY INDEX	INDEX1 0093
PS PSB	LEPSB	ERROR MESSAGE PSB	LEPSB *CEN
PS	ORDRPS00	ORDER PSB	ORDRPS 0059

*** END ***
0: C1 CH: LEB

PURCHASING MANAGEMENT SYSTEM			SG000008.LILI.CIV.1583			
LIST OF CODASYL SETS BY CODE						
SET	BLOCK	NAME OF THE BLOCK	LIN	OWN	MEM	LIBR.
SET001	DM4M1	TEST	300	MV00	MV01	0431
*** END ***						
0: C1 CH: LCC						

Database Blocks: Generation-Print

Lists and description reports on Database Blocks may be obtained by entering certain commands on-line on the Generation and Print Commands (GP) screen. The COMMANDS FOR PRINT REQUEST are listed below.

LISTS

LCB: List of all database blocks, sequenced by their codes.

C1 OPTION: Without explicit printed keywords,

C2 OPTION: With explicit printed keywords.

LEB: List of database blocks, sequenced by external names, without explicit printed keywords.

LTB: List of database blocks, sequenced by their types.

C1 OPTION: Without explicit printed keywords,

C2 OPTION: With explicit printed keywords.

LKB: List of all database blocks, by keywords.

After typing LKB, a selection field (SEL:) enables the user to choose implicit ('L') or explicit ('M') keywords, or both (' ').

Keywords are entered on a continuation line.

LTS: List of SQL objects by codes,

LES: List of SQL objects by external names.

DESCRIPTION

DTB: Definition, description and general documentation for the database block entered in the ENTITY CODE field. If no code is specified, ALL occurrences of the Database Block entity type are listed.

A Type selection is requested by entering the desired TYPE CODE field.

C1 OPTION: Provides definition, description, general documentation, and X-references,

C2 OPTION: With assigned text.

GENERATION OPTION

The Database Description Generation function is used to generate the specific DBMS source language according to the Database Block descriptions.

Chapter 6. Common Screens

Introduction

The screens presented in this chapter are common to several entities.

You will not find here detailed explanations specific to the input on each entity.

To find them, please consult the references to other Manuals located in each introduction text.

Error Messages - Help (-GE)

This screen enables you to describe error messages and help texts for Data Elements, Segments, Data Structures and Screens.

Refer to the 'OLSD' or 'Pacbench C/S - Business Logic and TUI Clients' manual, chapter 'Error Messages - Help Function', or to the 'Batch Applications' manual, chapter 'Error Messages'.

These manuals can be consulted from the following internet address:

<http://www.ibm.com/software/ad/vapabase/productinfo.htm>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'C'	Comment. See the 'Pacbench C/S Business Logic and TUI Clients' manual or the 'OLSD' manual, chapter 'Error Messages'.
			BATCH SYSTEMS DEVELOPMENT FUNCTION
		'S'	Replacement of an automatic error message
		'D'	Documentary messages. See the 'Batch Applications' manual, chapter 'Error Messages'.
4	60		DESCRIPTION
			BATCH SYSTEMS DEVELOPMENT FUNCTION
			DOCUMENTARY MESSAGE: 'D'-type line.
			Position 1:
			'0': Message before Element Description.
			'1': Message after Element Description.
			'2-5': Message after an error message of type 2 to 5.
			Position 2:
			blank.
			Position 3:
			'blank': Message entered on the line.
			'T': Call of a Text instance.
			Position 5-...: Documentary message, or Text & Paragraph codes (**: ALL paragraphs).
			OVERRIDING A STANDARD ERROR MESSAGE: 'S'-type line.
			Position 1:
			'2-5': Error type.
			Position 2: blank.
			Position 3: Error gravity (E, C or W).
			Position 4: Blank.
			Position 5-...: Error message.
			PACBENCH C/S AND ON-LINE SYSTEMS DEVELOPMENT FUNCTIONS
			SCREEN-RELATED DOCUMENTATION: 'C'- or 'T'-type line.
			Position 1-5: Blank.
			Position 6-...: Message on 'C'-type line or Text & Paragraph codes on 'T'-type line.
			OVERRIDING A DATA ELEMENT-RELATED STANDARD ERROR MESSAGE OR CREATING A DATA ELEMENT-RELATED USER-DEFINED ERROR MESSAGE:
			NOTE: Two lines need be coded.
			1. 'F'-type line:
			Position 1-6: Data Element instance code.
			2. 'U'-type line:
			Position 1-3: Blank.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Position 4:
		'2-5': Standard error type.
		'N' : Code of manual error (save '0' and '1')
		Position 5: Blank.
		Position 6-...: Error message.
		OVERRIDING STANDARD SEGMENT-RELATED ERROR MESSAGES: NOTE: Two lines need be coded.
		1. 'S'-type line:
		Position 1-4: Segment instance code.
		Position 5: Blank.
		Position 6: Category:
		'A' Heading,
		'R' Repetitive,
		'Z' Bottom.
		Position 7: Segment's rank in the category (if used several times in the category).
		2. 'U'-type line:
		Position 1:
		'F': Segment used in reception.
		'G': Segment used in display.
		Position 2-3: Blank.
		Position 4:
		'8': Invalid segment creation.
		'9': Segment not found.
		Position 5: Blank.
		Position 6-...: Error message.
		USER-DEFINED DIALOGUE-RELATED ERROR MESSAGE: 'U'-type line.
		Position 1-4: Error code.
		Position 5: Blank.
		Position 6-...: Error message.
		DOCUMENTATION OF A DATA ELEMENT:
		Position 1-3: Blank.
		Position 4: '0'(zero).
		Position 5: Blank.
		Position 6-...: Documentation lines, or Text and Paragraph codes after a 'T'-type line.
		DOCUMENTATION OF A DIALOGUE-RELATED ERROR MESSAGE:
		Position 1-4: Error code. 'blank' If following a 'U'-, 'C'-, or 'T'-type line.
		Position 5: Blank.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Position 6-...: Documentation lines, or Text and Paragraph codes after a 'T'-type line.
		DOCUMENTATION OF A STANDARD ERROR MESSAGE AND OF A USER-DEFINED DATA ELEMENT-RELATED ERROR MESSAGE:
		COL. VALUE DESCRIPTION
		Position 1-3: Error code.
		'blank' If following an 'F-', 'U-', 'C-', or 'T'-type line.
		Position 4: Type of error whose message is to be documented.
		Position 5: Blank.
		Position 6-...: Documentation lines, or Text and Paragraph codes after a 'T'-type line.
5	40	DESCRIPTION / SECOND PART
		You can tabulate directly to this field. However, it has not any special use in this screen.

Generation Elements (-GG)

You use this screen for different purposes, depending on the entity:

- on a Database Block Definition, you specify the physical characteristics of the Block.
See the Manual which corresponds to your Database Block Description manager.
- on the Database Block Description, you can complement the generation information.
See the Manual which corresponds to your Database Block Description manager.
- on a Data Structure, you enter technical information (frequency of backups...).
- on a Segment, you customize SQL accesses.
See the 'Relational Databases' Manual.
- on the Segment Call of Elements, you indicate additional description elements for the generation of Database Blocks.
See the 'Relational Databases' Manual.
- on a Screen or TUI Client Definition, you modify the generated constants (which depend on the hardware in use).
- for a TUI client only, you can generate one PSB per sub-monitor (as opposed to a general PSB for the whole dialogue). You must indicate, on the -GG screen of each sub-monitor, on a G-type line, in C2 option:
 - PSB= left-justified in the DESCRIPTION field
 - PSB code, at the next tabulation position.

This line must be the first -GG line.

NOTE: Each client component must belong to only ONE sub-monitor.

See the 'OLSD' and 'Pacbench C/S - Business Logic and TUI Clients' manuals.

The lines indicated in this screen are taken into account when the instance is generated.

All these manuals can be consulted at our internet address:

<http://www.ibm.com/software/ad/vapacbase/productinfo.htm>

CALLING A P.I.A.

The use of the Parameterized Input Aid entity (P.I.A.) may facilitate and standardize data entry on this screen.

An Input Aid is a group of parameterized lines. Each line contains a fixed label and a variable part which will be filled in the calling instance.

You call a P.I.A. by entering the value 'T' in the 'Type of Line' field and the P.I.A. code in the 'Description' field. VA Pac responds by displaying the P.I.A. lines field. Enter the parameters values on these formatted lines.

NOTE: The 'C2' option (O: C2) enables you to tab to the variable part of the P.I.A. line.

Additional lines cannot be inserted between called lines.

PREREQUISITES

The instance must be defined prior to being documented.

The called P.I.As must also exist.

LINES COPY

You may overwrite the instance code with another instance code (belonging to the same entity). This will not affect the original lines, but it will replicate them for the other instance.

NOTE: If the original lines include the lines of a called P.I.A., they will also be duplicated, except for user input on the variable parts of the P.I.A. lines.

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
	V	Generate, from a Segment, a customized SQL access without a delimiter.
	I	Call of a P.I.A.
		You can only call a P.I.A. with a 'G' type.
		The lines from the called P.I.A. have a 'Line generation option' set to 'blank' or 'G' in the P.I.A. Description.
		The call line is then replaced by the called P.I.A.
		Socrate Databases
	'V'	Line taken into account at generation.
	'P'	Line taken into account at generation.
	'Z'	Line taken into account at generation.
4	60	DESCRIPTION / FIRST PART
		If no P.I.A. is called, you can use the whole field.
		The contents of this field vary according to the type of line and to the current entity.
		See the corresponding manuals.
5	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.

Generation Options (-GO)

You use this screen for different purposes, depending on the entity:

- on an SQL Database Block, you specify options such as the prefixing, the transformation of lowercase into uppercase...
See the 'SQL Databases' Manual.
- on a Segment, you can inhibit the conversion of lowercase letters into uppercase letters on customized SQL accesses (with the UPPER=NO option).
See the 'SQL Databases' Manual.
- on a Program, you can change the function/subfunction number of the date transformation function (F9520 by default), with the DATPRO=ffss option.
See the 'Batch Applications' Manual.
- on a Library, you can add the function/subfunction (F9520) which transform the date, or enter the format of the generated indices.
See the 'Character-Mode User Interface' guide.
- on a Dialogue, you can indicate various generation options.
See the 'On-Line Systems Development' Manual.

- For all entities managed by the Pacbench C/S Function except blank-type or MW-type entities, you can indicate various generation options.
See the 'Business Logic' Volume of the Developer Documentation dedicated to eBusiness applications.

All these Manuals can be consulted at our internet address:

<http://www.ibm.com/software/ad/vapacbase/productinfo.htm>

CALLING A P.I.A.

The use of the Parameterized Input Aid entity (P.I.A.) may facilitate and standardize data entry on this screen.

An Input Aid is a group of parameterized lines. Each line contains a fixed label and a variable part which will be filled in the calling instance.

You call a P.I.A. by entering the value 'T' in the 'Type of Line' field and the P.I.A. code in the 'Description' field. VA Pac responds by displaying the P.I.A. lines field. Enter the parameters values on these formatted lines.

NOTE: The 'C2' option (O: C2) enables you to tab to the variable part of the P.I.A. line.

Additional lines cannot be inserted between called lines.

PREREQUISITES

The instance must be defined prior to being documented.

The called P.I.As must also exist.

LINES COPY

You may overwrite the instance code with another instance code (belonging to the same entity). This will not affect the original lines, but it will replicate them for the other instance.

NOTE: If the original lines include the lines of a called P.I.A., they will also be duplicated, except for user input on the variable parts of the P.I.A. lines.

NUM	LEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
			On an O-type line, Enter the option followed by an equal sign and the value (with no space).
5	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.

Chapter 7. Generation of Copy Books

Principles

The Specifications Dictionary includes a generator component. This component is used to obtain descriptions of Data Structures or SQL Database Blocks in COBOL source language, using the data structure descriptions implemented in the System.

Each description thus obtained is stored in a COBOL source library and can be incorporated into programs using the COPY clause.

A single data structure can be used to generate several different descriptions, each one adapted to a particular need in the programs. (i.e. in FILE SECTION or WORKING-STORAGE SECTION, taking the internal and input formats into account).

USAGE OF THE 'DATA' AND 'DATASQ' P.I.As.

The preliminary definition and description of the reserved P.I.A. called 'DATA' or 'DATASQ' is necessary for the generation of a data structure description.

At the data structure level, you can call the 'DATA' P.I.A. as often as necessary. Each time 'DATA' is called, you must specify the values in the parameters and the variants required for each description.

When you call the 'DATASQ' P.I.A., you request the description of an SQL Database Block. Two COPY clauses are generated for each Block: one for the host variables and one for the presence indicators.

The 'DATA' and 'DATASQ' P.I.As. are of Comment type ('C'). So they can be called only in the Comments screens (-GC) of a data structure.

The 'DATA' and 'DATASQ' P.I.As. are presented on the following pages.

There is one field per description line. The description must be scrupulously respected, be it the number, order or length of the parameters. No comment line is allowed.

Description of 'DATA' PIA

NUM	LEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
1	2		DATA STRUCTURE CODE IN GENER. DESCR.
		'A*'	Used to associate the 'DATA' P.I.A. to a COPY clause in order to regenerate the COPY clause in its initial format. See the TYPE field with value 'A' on the Data Element Definition screen, Data Element Description screen, and the General Documentation screen of the segment.
			default value : D.S. code
2	8		EXTERNAL NAME
			It is the file name in the program and in the COPY instruction.

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
3	1	DESCRIPTION LOCATION
	'F'	FILE SECTION
		The different structures of a record will not contain a REDEFINES clause, since VA Pac automatically generates all REDEFINES.
	'W'	WORKING-STORAGE SECTION or LINKAGE SECTION. The different record structures are redefined explicitly. (default value)
	'V'	Generation of a variable file.
4	1	TYPE OF COBOL TO GENERATE (REQUIRED)
		Specify the language variant to which the generated description must be adapted. This adaptation does not concern the USAGE clauses.
	'0'	Adaptation to ANSI COBOL: IBM MVS
	'1'	Adaptation to ANSI COBOL: IBM DOS
	'2'	Adaptation to ANSI COBOL: IBM 36
	'3'	Adaptation to COBOL : PC/MICROFOCUS
	'4'	Adaptation to COBOL : BULL DPS7
	'5'	Adaptation to ANSI COBOL: (74) BULL DPS8
	'6'	Adaptation to COBOL: (BCD) BULL DPS8
	'7'	Adaptation to COBOL: HP-3000
	'8'	This variant is required at the Library level to work in half-byte packed mode with UNISYS Series A or DPS8 hardware (values 5 and 8 for TYPE OF COBOL TO GENERATE on the Dialogue or Program definition).
		IMPORTANT NOTE: If this value is entered on the Li- brary Definition after data element formats have been defined, the element formats on the Element Definition and Segment/Screen Call of Elements, including FILLERS and undefined elements, will have to be re-entered so that the lengths are taken into account.
	'9'	Adaptation to ANSI COBOL: UNISYS 90/30
	'A'	Adaptation to COBOL: (74) PRIME
	'B'	Adaptation to COBOL: BURROUGHS (Medium systems),
	'D'	Adaptation to ANSI COBOL: (74) CONTROL DATA CORP.
	'E'	Adaptation to ANSI COBOL: (68) CONTROL DATA CORP.
	'F'	Adaptation to COBOL: TANDEM
	'T'	Adaptation to COBOL: DEC/VAX
	'J'	Adaptation to ANSI COBOL: PERKIN-ELMER-7-32
	'K'	Adaptation to ANSI COBOL: ICL 2900
	'M'	Adaptation to COBOL: DPS6
	'O'	Adaptation to COBOL: AS 400
	'R'	Adaptation to COBOL: IBM 34
	'S'	Adaptation to COBOL: SFENA

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
	'T'	Adaptation to ANSI COBOL: SIEMENS
	'U'	Adaptation to ANSI COBOL: (74) UNISYS 1100 Series
	'V'	Adaptation to ANSI COBOL: UNISYS 90/60
	'W'	Adaptation to COBOL: DPPX IBM 8100
	'X'	Adaptation to ANSI COBOL: IBM MVS VS COBOL II
	'Y'	Adaptation to COBOL: IBM 38
5	1	FORMAT TYPE
	'E'	Description using input formats.
	'I'	Description using internal formats with their associated 'usages'. (Default value).
	'S'	Description using output formats.
6	1	RECORD TYPE / USE WITHIN D.S.
		This option allows records of a data structure coming from the same description in a library to be arranged in several different ways:
	'blank'	Implicit or explicit redefinition of records. (Default value).
	'1'	Continuous sequence of records ('common part' segment followed by the different 'specific part' segments) without initial values or repetitions of records. If the data structure description appears in the COBOL FILE SECTION, the level number must be '2'.
	'2'	Continuous sequence of records that include initial values filled in on the description lines of the segments, or, by default, initial values of blank or zero according to the format (this option is reserved for descriptions in WORKING-STORAGE SECTION).
	'3'	Continuous sequence of records taking into account the number of repetitions specified on the Segment Definition screen. If the data structure description appears in the COBOL FILE SECTION, the level number must be '2'.
		This type of description can only be used for a data structure having a number of repetitions on the common part segment.
	'4'	Continuous sequence of records taking into account the number of repetitions specified on the Segment Definition screen. The associated level number must be '3'.
		The level '2' allows access to the table created by the repetition of a given record (FFEET).
		The level '1' consolidates all the information in the data structure (whether or not the common part and specific parts are repeated).
		This type of description can only be used for a data structure having a number of repetitions on the common part segment.
7	1	LEVEL NUMBER (COBOL) OF THE RECORD
		This option, used in relation with the preceding one, defines the level number of the descriptions of data structures, records, or data elements.
		In the following descriptions, the field 'FF00' is used to define the data structure level.

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
	'1'	Level '01' for data structures and records (default option). If the data structure description appears in the COBOL FILE SECTION, the records must be redefined. If the data structure has no 'common part' with a RECORD TYPE / USE WITHIN D.S. other than 'blank', the file level does not appear.
	'2'	Level '01' for data structures and '02' for records, the '01' level does not appear if the RECORD TYPE is 'blank'.
	'3'	Level '02' for data structures and '03' for records, when associated with RECORD TYPES 1, 2 or 3. Level '01' for data structures and '03' for records, when associated with RECORD TYPE 4. Level '03' for data structures and records when associated with RECORD TYPE 'blank'.
8	2	CONTROL CARDS FRONT/BACK COPY BOOK
		This field represents the two options of Job Control Cards which are used to ensure that the generated description is catalogued in the source library:
		.the option code to be inserted into the control cards in front of a generated description,
		.the option code to be inserted into the control cards in back of each generated description.
9	40	SEGMENT SELECTION
		If there is no selection, all the segments making up a data structure are selected.
		When a specific selection is made, the sequence of the chosen record codes, (which may or may not be sorted) is called for.
		The selection of a 'common part' segment of a multi- record file must be explicitly specified.

Description of 'DATASQ' PIA

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
1	2	DATA STRUCTURE CODE IN GENER. DESCR.
	'A*'	Used to associate the 'DATA' P.I.A. to a COPY clause in order to regenerate the COPY clause in its initial format. See the TYPE field with value 'A' on the Data Element Definition screen, Data Element Description screen, and the General Documentation screen of the segment.
		default value : D.S. code
2	8	EXTERNAL NAME
		It is the file name in the program and in the COPY instruction.
3	6	BLOCK CODE
		One to six alphanumeric characters.
4	1	TYPE OF COBOL TO GENERATE (REQUIRED)
		Specify the language variant to which the generated description must be adapted. This adaptation does not concern the USAGE clauses.

NUM	LEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
		'0'	Adaptation to ANSI COBOL: IBM MVS
		'1'	Adaptation to ANSI COBOL: IBM DOS
		'2'	Adaptation to ANSI COBOL: IBM 36
		'3'	Adaptation to COBOL : PC/MICROFOCUS
		'4'	Adaptation to COBOL : BULL DPS7
		'5'	Adaptation to ANSI COBOL: (74) BULL DPS8
		'6'	Adaptation to COBOL: (BCD) BULL DPS8
		'7'	Adaptation to COBOL: HP-3000
		'8'	This variant is required at the Library level to work in half-byte packed mode with UNISYS Series A or DPS8 hardware (values 5 and 8 for TYPE OF COBOL TO GENERATE on the Dialogue or Program definition).
			IMPORTANT NOTE: If this value is entered on the Library Definition after data element formats have been defined, the element formats on the Element Definition and Segment/Screen Call of Elements, including FILLERS and undefined elements, will have to be re-entered so that the lengths are taken into account.
		'9'	Adaptation to ANSI COBOL: UNISYS 90/30
		'A'	Adaptation to COBOL: (74) PRIME
		'B'	Adaptation to COBOL: BURROUGHS (Medium systems),
		'D'	Adaptation to ANSI COBOL: (74) CONTROL DATA CORP.
		'E'	Adaptation to ANSI COBOL: (68) CONTROL DATA CORP.
		'F'	Adaptation to COBOL: TANDEM
		'I'	Adaptation to COBOL: DEC/VAX
		'J'	Adaptation to ANSI COBOL: PERKIN-ELMER-7-32
		'K'	Adaptation to ANSI COBOL: ICL 2900
		'M'	Adaptation to COBOL: DPS6
		'O'	Adaptation to COBOL: AS 400
		'R'	Adaptation to COBOL: IBM 34
		'S'	Adaptation to COBOL: SFENA
		'T'	Adaptation to ANSI COBOL: SIEMENS
		'U'	Adaptation to ANSI COBOL: (74) UNISYS 1100 Series
		'V'	Adaptation to ANSI COBOL: UNISYS 90/60
		'W'	Adaptation to COBOL: DPPX IBM 8100
		'X'	Adaptation to ANSI COBOL: IBM MVS VS COBOL II
		'Y'	Adaptation to COBOL: IBM 38
5	1		GENERATION OF PRESENCE INDICATORS
		Y	Presence indicators will be generated.
		N	Presence indicators will not be generated.
6	2		CONTROL CARDS FRONT/BACK COPY BOOK

NUMLEN	VALUE CLASS	AND FILLING MODE DESCRIPTION OF FIELDS
		This field represents the two options of Job Control Cards which are used to ensure that the generated description is catalogued in the source library:
		.the option code to be inserted into the control cards in front of a generated description,
		.the option code to be inserted into the control cards in back of each generated description.
7	40	SEGMENT SELECTION
		If there is no selection, all the segments making up a data structure are selected.
		When a specific selection is made, the sequence of the chosen record codes, (which may or may not be sorted) is called for.
		The selection of a 'common part' segment of a multi- record file must be explicitly specified.

Data Generation and/or Printing

To generate data descriptions from a Data Structure, you must:

- call the 'DATA' or 'DATASQ' P.I.A. in the Comments screen (-GC) of the Data Structure,
- valorize the parameters,
- request the generation on the generation and printing screen (CH: GP).
You request it via the 'GCD' Command, followed by the Data Structure code.
You can also enter optional commands (CCF:_ and CCB:_ fields).

General 'Flow' generation information (such as name of the source library..) may be coded on a 'FLD' Command.

With the C3 option, the 'GCD' command enables you to generate, besides the Segment Cobol source, Data Element and Segment description lines in the form of comments which will be used by VAP Connector (an eBusiness tool).

The 'GCD' command also enables you to valorize the parameters of the DATA and DATASQ P.I.As. on continuation lines:

- In C1, C2 or C3 option, these parameters' values entered on continuation lines override the values entered in ALL the calls to the DATA and DATASQ P.I.As.
- In C4 option, the P.I.A. calls, if any, are ignored. The parameters entered on continuation lines are optional (because they all have a default value), except the SQLBLO parameter which is mandatory if the PIATYP parameter equals DSQC1 or DSQC2.

The following table presents the parameters you enter on continuation lines in order to override the values indicated in the DATA P.I.A. and/or the DATASQ P.I.A..

Param code	Meaning	Length	Default value	DATA	DATASQ
PIATYP °	P.I.A. type	3 or 5	DC1 for DATA C1 °°	X	X
DSCODE	Data Structure code	2	code of generated DS	X	X

Param code	Meaning	Length	Default value	DATA	DATASQ
EXTNAM	External name	8	code of generated DS	X	X
DESLOC	Description location	1	W	X	
COBTYP	Variant	1	space	X	
			0		X
FORTYP	format	1	I	X	
RECTYP	Description type	1	space	X	
LEVNUM	Description level	1	1	X	
SEGSEL	Record selection	40	space	X	X
SQLBLO	Block code, MANDATORY if PIATYP=DSQC1 or DSQC2	6	block code		X
PREIND	Presence indicator	1	Y		X

°: only for the C4 option.

°°: Other values are possible (DC2 for DATA C2, DC3 for DATA C3, DSQC1 for DATASQ C1 and DSQC2 for DATASQ C2).

Example of Generated 'DATA' in the FILE SECTION

```

05          TR00-00.
10          TR00-KEYCD.
11          TR00-COCARA PICTURE X.
11          TR00-NUCOM PICTURE 9(5).
11          TR00-FOURNI PICTURE X(3).
05          TR00-SUITE.
15          FILLER          PICTURE X(00157).
01          TR05.
10          FILLER          PICTURE X(00009).
10          TR05-NUCLIE PICTURE 9(8).
10          TR05-DATE PICTURE X(6).
10          TR05-RELEA PICTURE X(3).
10          TR05-REFCLI PICTURE X(30).
10          TR05-RUE PICTURE X(40).
10          TR05-COPOS PICTURE X(5).
10          TR05-VILLE PICTURE X(20).
10          TR05-CORRES PICTURE X(25).
10          TR05-REMIS PICTURE S9(4)V99.
10          TR05-MATE PICTURE X(8).
10          TR05-LANGU PICTURE X.
10          TR05-FILLER PICTURE X(5).
01          TR10.
10          FILLER          PICTURE X(00009).
10          TR10-QTMAC PICTURE 99.
10          TR10-QTMAL PICTURE 99.
10          TR10-INFOR PICTURE X(35).
10          TR10-RFI PICTURE X(100).
10          FILLER          PICTURE X(00018).
01          TR20.
10          FILLER          PICTURE X(00009).
10          TR20-EDIT PICTURE X.
10          FILLER          PICTURE X(00156).
01          TR30.
10          FILLER          PICTURE X(00009).
10          TR30-NUCOM PICTURE 9(5).
10          TR30-LV00.

```

```

11          TR30-NOCL.
12          TR30-NOCL11 PICTURE X.
12          TR30-NOCL12 PICTURE XX.
12          TR30-NOCL2 PICTURE XX.
11          TR30-NBLIV PICTURE 9.
11          TR30-QTLI PICTURE S9(5)V99
            COMPUTATIONAL-3.
11          TR30-GROUPE
            OCCURS      009
01          TR00.
            DEPENDING ON      TR00-NBLIV.
12          TR30-DALI PICTURE X(6).
12          TR30-QULI PICTURE S9(5)V99
            COMPUTATIONAL-3.

```

Example of Generated 'DATA' in WORKING-STORAGE

```

01          G-AT20.
04          G-AT20-PARAM.
10          G-AT20-LOZTR PICTURE S9(4) COMPUTATIONAL
            VALUE +025.
10          G-AT20-ADRCLC PICTURE S9(4) COMPUTATIONAL
            VALUE +001.
10          G-AT20-LOCLE PICTURE S9(4) COMPUTATIONAL
            VALUE +005.
10          G-AT20-NUAPP PICTURE 99
            VALUE ZERO.
10          G-AT20-NUTAB PICTURE X(6)
            VALUE 'POSTES'.
10          G-AT20-TABFO PICTURE XX VALUE SPACE.
10          G-AT20-TABCR PICTURE XX VALUE SPACE.
10          G-AT20-DAHTA PICTURE X(6) VALUE SPACE.
10          G-AT20-NUSSC PICTURE X VALUE '1'.
10          G-AT20-NUSSY PICTURE X VALUE SPACE.
10          G-AT20-TRANID PICTURE X(4) VALUE SPACE.
10          G-AT20-FILSYS PICTURE X(30) VALUE SPACE.
04          AT20.
10          AT20-COPOS.
15          AT20-CODEPA PICTURE XX
            VALUE SPACE.
15          AT20-COCOM PICTURE X(3)
            VALUE SPACE.
10          AT20-VILLE PICTURE X(20)
            VALUE SPACE.
01          CD00.
10          CD00-KEYCD.
15          CD00-COCARA PICTURE X.
15          CD00-NUCOM PICTURE 9(5).
15          CD00-FOURNI PICTURE X(3).
10          CD00-SUITE.
15          FILLER PICTURE X(00157).
01          CD05 REDEFINES CD00.
10          FILLER PICTURE X(00009).
WORKING-STORAGE SECTION.
10          CD05-NUCLIE PICTURE 9(8).
10          ME00-MESSA PICTURE X(75).
10          CD05-DATE PICTURE X(6).
10          CD05-RELEA PICTURE X(3).
10          CD05-REFCLI PICTURE X(30).
10          CD05-RUE PICTURE X(40).
10          CD05-COPOS PICTURE X(5).
10          CD05-VILLE PICTURE X(20).
10          CD05-CORRES PICTURE X(25).
10          CD05-REMIS PICTURE S9(4)V99.
10          CD05-MATE PICTURE X(8).
10          CD05-LANGU PICTURE X.
10          CD05-FILLER PICTURE X(5).

```



```

01          CD10      REDEFINES  CD00.
          10          FILLER      PICTURE X(00009).
          10          CD10-QTMAC  PICTURE 99.
          10          CD10-QTMAL  PICTURE 99.
          10          CD10-INFOR  PICTURE X(35).
          10          CD10-RFI    PICTURE X(100).
          10          FILLER      PICTURE X(00018).
01          CD20      REDEFINES  CD00.
          10          FILLER      PICTURE X(00009).
          10          CD20-EDIT  PICTURE X.
          10          FILLER      PICTURE X(00156).
01          F010.
          10          F010-CLEFO.
          15          F010-FOURNI PICTURE X(3).
          15          F010-MATE  PICTURE X(8).
          15          F010-RELEA PICTURE X(3).
          15          F010-LANGU PICTURE X.
          15          F010-FILLER PICTURE X(5).
          10          F010-QTMAS  PICTURE 9(4).
          10          F010-QTMAM  PICTURE 9(4).
          10          F010-LIBFO  PICTURE X(20).
          10          F010-FILLER PICTURE XX.
01          ME00.
          10          ME00-CLEME.
          15          ME00-COPERS PICTURE X(5).
          15          ME00-NUMORD PICTURE XX.

```




Part Number: DDSPE000351A - 6618

Printed in USA