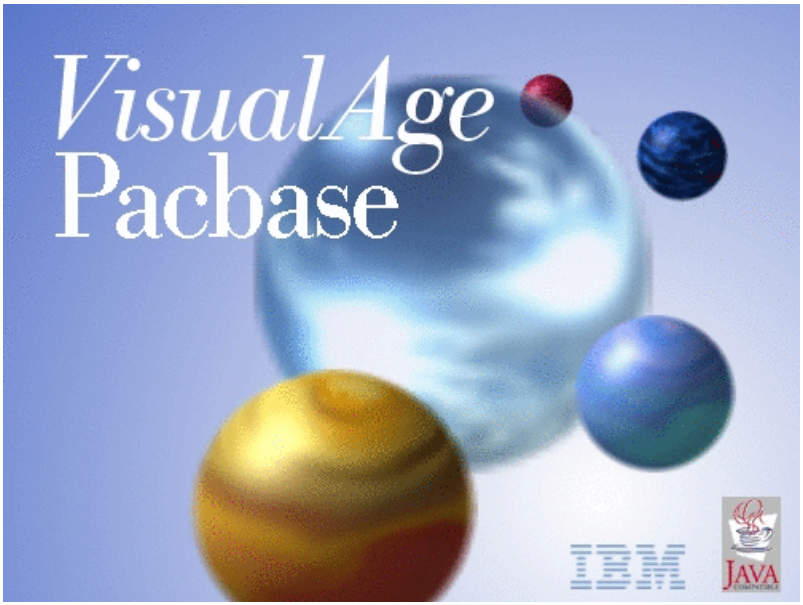
VisualAge Pacbase

RELATIONAL/SQL DATABASE DESCRIPTION

Version 3.5



VisualAge Pacbase

RELATIONAL/SQL DATABASE DESCRIPTION

Version 3.5

Note

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

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

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

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

First Edition (September 2006)

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

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

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

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

Contents

Notices

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

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

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

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

Trademarks

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

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

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

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

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

Chapter 1. Introduction

Introduction to the Database Description Function

The Database Description function automatically generates database descriptions adapted to the database management system in use. This is done by using segment and relationship descriptions defined during the application analysis phase.

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

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

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

DBD/RELATIONAL SQL

This function can only be used in conjunction with the Dictionary: data defined in the Specifications Dictionary (whether or not the METHODOLOGY function is being used) can be used to generate database descriptions.

This information is described through a database description language which is independent of the DBMS in use. This allows the user to generate different descriptions from the same source.

Principles of Description

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

- An introductory comment explaining the purpose and the general characteristics of the entity or screen,
- A detailed description of each screen, including the input fields for on-line screens data entry into the Database.

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

All on-line fields described in this manual are assigned an order number. These numbers are displayed on the screen examples which appear before the input field descriptions and allow for easy identification of a given field.

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

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

NOTE: Each type of Database Block has a specific description. However, several Database Block types may use the same Batch Form.

As a result, fields may have different meanings or may not be used, depending on the type of Database Block.

Chapter 2. Use of the Function with SQL

Introduction

The Relational/SQL DBD function is used to manage relational/SQL databases with the SQL language and to generate (in on-line or batch mode) the Data Description Language (DDL) of the database block in order to create, alter or delete TABLESPACES, TABLES, VIEWS and INDEXES, using the information contained in the Database.

The Relational/SQL DBD Manual is not a technical training manual for database management.

In particular, you should be familiar with the SQL Data Description Language (DDL) and the Specifications Dictionary.

The purpose of this manual is to guide you through the description of a relational database in the Specifications Dictionary.

Use of Entities

When related to the DBD Description function, the Specifications Dictionary manages logical descriptions of the various external views which will be processed by programs.

The following entities are used:

.Data Elements,

.Segments,

.Database Blocks,

.Generation Elements and Options, Comments of Segments and Database Blocks, as well as Error Messages of Segments.

.Parameterized Input Aids.

TERMINOLOGICAL EQUIVALENTS BETWEEN RELATIONAL DBMS AND THE SYSTEM'S METAMODEL

A relational database is a set of physical environments (TABLESPACE, DBSPACE, SPACE, STORAGE-AREA, etc.; these environments will be referred as SPACES in this manual).

A Space is a set of Tables. A Table is a set of Columns.

A Database Block is a set of Segments; a Segment is a set of Data Elements.

This parallelism is illustrated in the following chart:

SQL OBJECT	EQUIVALENT
DATABASE	DATABASE BLOCK (TYPE OF BLOCK = 'Qx')
SPACE	BLOCK DESCRIPTION (-DR): 'P' TYPE LINE
TABLE	SEGMENT ON A 'T' TYPE LINE OR AN 'A' TYPE LINE FOR AN 'ALTER TABLE' (-DR)
VIEW	SEGMENT ON A 'V' TYPE LINE (-DR SCREEN)
COLUMN	DATA ELEMENT
INDEX	DATA ELEMENT OF A SEGMENT CALLED ON AN 'I' TYPE LINE (-DR SCREEN)
PRIMARY KEY	DATA ELEMENT OF A SEGMENT CALLED ON A 'K' TYPE LINE (-DR SCREEN)
FOREIGN KEY	DATA ELEMENT OF A SEGMENT CALLED ON A 'J' TYPE LINE (-DR SCREEN)
PACKAGE	DATA ELEMENT OF A SEGMENT CALLED ON A 'C' LINE (-DR SCREEN)
FUNCTION	DATA ELEMENT OF A SEGMENT CALLED ON A 'E' LINE (-DR SCREEN)
PROCEDURE	DATA ELEMENT OF A SEGMENT CALLED ON A 'Q' LINE (-DR SCREEN)
TRIGGER OR RULE	DATA ELEMENT OF A SEGMENT CALLED ON A 'R' LINE (-DR SCREEN)

GENERATION OF A DATABASE

A database is generated from a Database Block.

Taking into account all of the information in the Specifications Dictionary (logical level information) and according to the type of block, the DBD function ensures the generation of the Data Description Language (DDL); and it also generates the DDL for the Tables, Views, and Indexes according to the information provided on the Segment Description and Data Element Definition screens.

You may replace or complete the generated lines, using the Block's Generation Elements (-GG) lines, Generation Options (-GO) lines and Comments (-GC) lines.

IMPLEMENTATION

SQL commands are generated according to the input on the Database Block Description (-DR) screen, which includes the segment calls (which contain the descriptions of Tables and Views).

The Data Elements called in the Segments are used to generate the Index descriptions.

Tablespaces are also defined on this screen; they are described on Generation Elements (-GG) lines associated with the block description lines (-DRnnnGG, where 'nnn' = the description line number).

NOTE: Tablespaces are not generated for INTEREL RDBC, INTEREL RFM, NON STOP SQL, DB2/2, DB2/6000, SYBASE and SQL SERVER.

The following screens depend on the input on the Database Block Description (-DR) screen:

- Input of Data Element of Key (-DRnnnK) screen, used to consult and update the Primary and Foreign keys and to select the columns to be added or modified in a table.
- SQL Commands Generation (-GNnnn) screen, used to enter the information necessary for the generation of SQL commands (for ex., selection of the Columns to be added to a Table); the '-GNnnn' screen can also be accessed by entering a 'Q' in the ACTION CODE field of an 'nnn' description line on the -DR screen.
- Display of generated SQL commands once the 'ENTER' key has been pressed on the '-GN' screen.

On Generation Elements (-GG), Generation Options (-GO) and Comments (-GC) screens, virtual lines and P.I.A.'s facilitate user input.

Chapter 3. Column: Data Element

Column Definition (E.....)

Since a Column corresponds to a Data ELement, it must be defined on a Data Element Definition screen accessed via the following CHOICE:

CH: E.....

ASSOCIATED LINES

Comments (-GC).

COLUMN NAME

A Column Name may be entered on the Segment Call of Elements (-CE) screen (see Chapter ″TABLE OR VIEW: SEGMENT″, Subchapter ″TABLE OR VIEW DESCRIPTION″), or on an ’R’-type line on the Data Element Description (-D) screen (for further details please refer to Subchapter ″COLUMN DESCRIPTION″).

If a Column Name is entered on both screens, the one entered on the Segment Call of Elements (-CE) screen is the one taken into account.

If no Column Name is specified, the Data Element code will be used as the Column Name.

COLUMN FORMATS

The System generates the Column’s format (type and length) from input in the INTERNAL FORMAT and USAGE fields on the Data Element Definition screen.

The following charts show the generated formats in relation to the formats entered on the Data Element’s Definition screen for each type of database.

A CHAR(n) is generated for the System date formats (M, G, D, I, E, S, C, T, TS) which do not generate an SQL date format.

If you input a non-standard date format on the -D screen of the Data Element (on an E-type line), a DATE format is generated for ORACLE, SYBASE and SQL SERVER.

DB2, SQL/DS

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n) with n<255	D	
VARCHAR(n)	X(n) with n<255	D	(see below)
LONG VARCHAR	X(n) with n>254	D	(see below)
SMALLINT	S9(n) with n<5	C	
INTEGER	S9(n) with n>4	C	
DECIMAL(n+m,m)	S9(n)V9(m)	3,1	use 1 for DB2 only
FLOAT	No format	W	
DATE	M, G	D	
TIME	T	D	
TIMESTAMP	TS	D	
REAL	No format	F	DB2 only

DB2/2, DB2/6000

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	
VARCHAR(n)	X(n) with n<4001	D	(see below)
LONG VARCHAR	X(n) with n>4000	D	(see below)
SMALLINT	S9(n) with n<5	C or G	
INTEGER	S9(n) with n>4	C or G	
DECIMAL(n+m,m)	S9(n)V9(m)	3,1	
FLOAT	No format	W	
DATE	M, G	D	
TIME	T	D	
TIMESTAMP	TS	D	

ORACLE < V7

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	V5: n<241 V6: n<2001
LONG VARCHAR	X(n)	D	n<65536 (see below)
INTEGER	S9(n)	C,J,P,5,6	n<=4
NUMBER(n+m,m)	S9(n)V9(m)	T,Q,3,8,9	

SQL FORMAT	FORMAT	USE	NOTES
FLOAT	No format	W	
DATE	D, I, E, S, C, M, G, T	D	

RDMS 1100

SQL FORMAT	FORMAT	USE
CHAR(n)	X(n)	D
DECIMAL (n+m+1,m)	S9(n)V9(m)	D,C,3,H

FROM VERSION 5RA4 ONWARDS

SQL FORMAT	FORMAT	USE
CHAR(n)	X(n)	D
DECIMAL (n+m+1,m)	S9(n)V9(m)	3, C
NUMERIC (n+m,m)	S9(n)V9(m) 9(n)V9(m)	H or D
FLOAT	No format	F COMP-1
DOUBLE PRECISION	No format	W COMP-2

ORACLE V7

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	n<2000 and key indic= 'C'
VARCHAR2(n)	X(n)	D	n<2000
VARCHAR(n)	X(n)	D	n<2000 and key indic='V'or'W'
LONG VARCHAR	X(n)	D	n<65536 and key indic= 'L'
INTEGER	S9(n)	C,J,P,5,6	
NUMBER(n+m,m)	S9(n)V9(m)	T,Q,3,8,9	
FLOAT	No format	W	
DATE	D, I, E, S, C, M, G, T	D	

DATACOM/DB

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	with n<32768

SQL FORMAT	FORMAT	USE	NOTES
NUMERIC(n+m,n)	S9(n)V9(m)	D	
DECIMAL(n+m,n)	S9(n)V9(m)	3	
INTEGER	S9(n) 4<n<9	C	
SMALLINT	S9(n) with n<5	C	
DOUBLE PRECISION	No format	W	
DATE	M	D	
TIME	T	D	
TIMESTAMP	TS	D	

NON STOP SQL

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	n<4075
VARCHAR(n)	X(n)	D	V in -CE (see below)
DECIMAL(n+m,n)	S9(n)V9(m)	D, 1	
DECIMAL(n+m,n) Unsigned	9(n)V9(m)		
NUMERIC(n+m,n)	S9(n)V9(m)	C	
NUMERIC(n+m,n) Unsigned	9(n)V9(m)		
SMALLINT	S9(n)	C	n<5
SMALLINT Unsigned	9(n)		
INTEGER	S9(n)	C	n<10
INTEGER Unsigned	9(n)		
LARGEINT	S9(n)	C	n<19
DATE	M	D	Version C30
TIME	T	D	Version C30
TIMESTAMP	TS	D	Version C30

INTEREL RDBC

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	
VARCHAR(n)	X(n) with n<32000	D	V on '-CE' (see below)
LONG VARCHAR	X(n) with n=32000	D	V on '-CE' (see below)

SQL FORMAT	FORMAT	USE	NOTES
SMALLINT	no format	0 (zero)	
INTEGER	no format	J	
DECIMAL(n+m,m)	S9(n)V9(m)	D,1	
REAL	No format	F	
DATE	E, M	D	
FLOAT	No format	W	

INTEREL RFM

SQL FORMAT	FORMAT	USE	NOTES
CHAR(n)	X(n)	D	n<4096
SMALLINT	S9(n)	0 (zero)	n<5
INTEGER(n)	S9(n)	J	4<n<10
DECIMAL(n+m,m)	S9(n)V9(m)	D	
FLOAT	-	W	
REAL	-	F	
DATETIME YEAR TO DAY	G	D	8 charac. format
TIME	T	D	
TIMESTAMP	TS	D	

SYBASE

SQL FORMAT	FORMAT	USAGE	NOTES
CHAR(n)	X(n)	D	n<255
VARCHAR(n)	X(n)	D	n<255
SMALLINT	S9(n) or 9(n)	C, G	n<5
INTEGER	S9(n) or 9(n)	C, G	n<9
NUMERIC(m,n)	S9(m-n)V9(n) or 9(m-n)V9(n)	D,3,1,X	
DECIMAL(n,m)	S9(n)V9(m)	C	
REAL	No format	F	
FLOAT	No format	W	
DATETIME	T, M, S, E, I, G	D-dff	

SQL SERVER

SQL FORMAT	FORMAT	USAGE	NOTES
CHAR(n)	X(n)	D	n<255
VARCHAR(n)	X(n)	D	n<255
SMALLINT	S9(n) or 9(n)	C, G	n<5
INTEGER	S9(n) or 9(n)	C, G	n<9
NUMERIC(m,n)	S9(m-n)V9(n) or 9(m-n)V9(n)	D,3,1,X	
DECIMAL(n,m)	S9(n)V9(m)	C	
REAL	No format	F	
FLOAT	No format	W	
DATETIME	T, M, S, E, I, G	D-dff	

VARIABLE COLUMNS

In order to generate a variable Column, you must define a data element with an alphanumeric format X(n). On the data element call line on the Table Segment Call of Elements (-CE) screen, you indicate that it is a variable length field by entering a 'V' in the KEY INDICATOR FOR ACCESS OR SORT field.

The System generates:

.VARCHAR(n)

NOTE: This type of column is only generated for DB2, SQL/DS, INTEREL RDBC, NONSTOP SQL, DB2/2, DB2/6000, SYBASE and SQL SERVER.

For DB2 and SQL/DS, if 'n' is greater than 254, the System generates:

.LONG VARCHAR.

For DB2/2 and DB2/6000, if 'n' is greater than 4000, the System generates:

.LONG VARCHAR.

Note that in the programs generated with the BATCH S.D. and ON-LINE S.D. functions, the corresponding 'Host Variables' are processed as variable alphanumeric strings.

RDMS 1100, DATACOM/DB:

Variable columns cannot be generated.

DB2, SQL/DS, DB2/2 and DB2/6000:

- value 'W' generates a VARCHAR,
- value 'L' generates a LONG VARCHAR.

ORACLE V5 and V6:

- value 'V' only generates a VARCHAR.
- value 'L' generates a LONG VARCHAR.
- value 'W' generates a VARCHAR.

ORACLE V7:

- value 'V' or 'W' generates a VARCHAR(n).
- field length<2000 and 'space' in key indicator generate a VARCHAR2(n).
- value 'L' generates a LONG VARCHAR.
- value 'C' generates a CHAR.

REQUIRED COLUMNS

On a Data Element call line on the Table Segment's '-CE' screen, you can specify a required Column (NOT NULL), or a required Column with default values, as follows:

In the PRESENCE INDICATOR field, enter:

```
'O' --> NOT NULL,
'P' --> NOT NULL WITH DEFAULT (for DB2,
        DATACOM/DB, DB2/2 and DB2/6000)
        DEFAULT               (for INTEREL RDBC)
        DEFAULT SYSTEM        (for NONSTOP SQL C30)
```

SQL/DS, ORACLE, NONSTOP SQL, SYBASE:

It is not possible to generate a (NOT NULL WITH) DEFAULT clause. Value 'P' is not recognized.

RDMS 1100:

It is not possible to generate required Columns (the value in the PRESENCE INDICATOR field is ignored).

SQL SERVER:

In the PRESENCE INDICATOR field, enter:

```
'O' --> NOT NULL
```

```
'P' --> NOT NULL DEFAULT SPACE (nn) for an alphanumeric
        field.
    --> NOT NULL DEFAULT 0 for a numeric field.
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260

DATA ELEMENT CODE   1 REFCLI

NAME..............:2  CUSTOMER REFERENCES
TYPE..............:3  R

INPUT FORMAT.......:5  X(30)                                  LENGTH...:    30
INTERNAL FORMAT....:6  X(30)                  USAGE : 7 D     LENGTH...:    30
OUTPUT FORMAT......:8  X(30)                       Z: 9       LENGTH...:    30

EXPLICIT KEYWORDS..: 10

PARENT ELEMENT.....: 11









SESSION NUMBER.....: 0067        LIBRARY......: BMS    LOCK....:


O: C1 CH: Erefcli                      ACTION:
```

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

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

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

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

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

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

Column Description (E.....D)

A Column is described on a Data Element Description (-D) screen which is accessed via the following CHOICE:

CH: E.....D

COLUMN NAME

By default, the Column Name is the 6-character code of the data element. This default may be overridden by entering an 18-character Column Name on the Data Element Description (-D) screen with a TYPE OF LINE = 'R'.

The Column Name may also be entered on the Segment Call of Elements screen; in this case, it overrides any name entered on the Data Element Description (-D).

PREREQUISITE

The data element must have been previously defined.

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
 ELEMENT DESCRIPTION           REFCLI CUSTOMER REFERENCE1
2 3     4 5 6          7
A LIN : T S VALUE      SIGNIFICANCE - DESCRIPTION
  100 : L              CUST. REF.         ***   SQL NAME   ***
  150 : P              LP-KCP  ORDER NUMBER: 05179
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: E refcli D
```

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
4	1		Type of line
		'blank'	Value and/or description line.
			With a blank line type, descriptive text is assigned to the Data Element. This text includes all possible values and what they mean.
		'D'	DATA ELEMENT DEFAULT VALUE
			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 '£'.
		'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.

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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:
		'+'	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 'I':
		'C'	Creation.
		'M'	Modification.
		'D'	Deletion.
		'X'	Mod-4 (implicit update).
			WITH TYPE OF LINE 'O':

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

NUM	LEN	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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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:
			. 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).

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

Chapter 4. Table or View: Segment

Table or View Definition (S....)

A Table or View being similar to a Segment, it is defined on a Segment Definition screen, accessed via the CHOICE:

CH: S....

You specify that the Segment is a Table or View when calling it into the database block.

The external name of the Table or View may be entered in the CODE/VALUE OF RECORD TYPE ELEMENT field (between quotes) on the Segment Definition Screen.

The default for the external name is the SEGMENT CODE.

ASSOCIATED LINES

Comments (choice CH: -GC).

Generation Elements (choice CH: -GG).

Generation Options (choice CH: -GO).

Error Messages-Help (choice CH: -GE).

PREREQUISITE

The Data Structure must have been previously defined.

TABLE OR VIEW NAME PREFIXING

The name of a Table or View may be generated from different user input. They are taken into account by the DBD Description function in the following order of priority:

1. 27-character name entered on the Table or View segment call line in the block. For more information, you should refer to subchapter ″TABLE OR VIEW DESCRIPTION″.
2. 8-character name entered on the Segment Definition screen, in the CODE / VALUE OF RECORD TYPE ELEMENT field.

3. When no external name is specified, the assigned name is the Segment code.

DB2, SQL/DS, DB2/2 AND DB2/6000: PREFIXING MODE

The presence of a period within the name conditions the prefixing mode:

- If the name is formatted as follows:

  PREFIX.NAME

  it will processed as such by the System and the DBMS i.e. as explicitly prefixed by the user.
- If the name contains neither a period nor a prefix, the DBD function prefixes the name with the user's 8-character User Code. The DBMS will process it as such, i.e. as explicitly prefixed by the System.
- If the name is formatted as follows:

  .NAME

  the DBD function generates it without the period so that the DBMS ensures prefixing. In this case, the assigned prefix depends on the catalog update mode:
  - Within a TSO environment, via the interactive SQL application (SPUFI), the TSO LOGON is used as the prefix. In case of batch updating, the prefix is a parameter in the JOB.
  - Within a CICS or IMS environment, via the interactive application (-GN), the prefix depends on the SIGNON procedure and also on whether or not a Security System is in operation.

    EXAMPLES:
  - Under CICS with CSSN SIGNON, the prefix is the OPID associated with the USERID in the DFHSNT.
  - Under IMS with RACF and LOGON input, the prefix is the IMS LOGON.

NOTE: You can modify the prefixing mode by entering PREFIX = NO on an 'O'-type line in the 'Generation Options' screen (B.....GO). Then a name which contains neither a period nor a prefix will be generated as such, and a .NAME name will be generated prefixed by the user code.

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
                             1 2
SEGMENT DEFINITION........: DZ05

NAME.....................:3 FOURNITURE

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



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


EXPLICIT KEYWORDS..:8


SESSION NUMBER.....: 0067        LIBRARY......: BMS    LOCK....:


O: C1 CH: S dz05                      ACTION:
```

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

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

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

Table or View Description (S....CE)

Since a Table or View is defined as a segment, it is described on the Segment Call of Elements Screen. This screen is accessed via the following input in the CHOICE field:

CH: S....CE

A Table or View description is composed of the list of its Columns. Since Columns are represented by Data Elements, the corresponding Data Elements are called into the segment.

Segments may also be called into Segments describing Relational Tables and Views.

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

PREREQUISITES

The Table or View and its Columns must have been previously defined.

ASSOCIATED LINES

Comments associated with each line (choice CH: -CEnnnGC).

Generation Elements associated with each line (choice CH: -CEnnnGG).

Error Messages - Help associated with each line (choice CH: -CEnnnGE).

where 'nnn' is the number of the call line.

NOTE ON VIEW SOURCE DESCRIPTIONS

The DB2 View Description (-DBE) screen enables you to specify the source (tables) of the Elements (Columns) called into Segments used as Views. This screen is accessed by entering '-DBE' in the CHOICE field from the Segment screen network. Once entered, these lines will be displayed on the Call of Elements Screen with "(-DBE)" as the DATA ELEMENT CODE and an asterisk (*) in the ACTION CODE field.

For more information, see Subchapter 'ADDITIONAL VIEW DESCRIPTION'.

NOTE: Data Elements which are used as group Elements or redefined, are not taken into account for the generation. A Data Element may be called only once in each Table or View, even if it is given different Column Names.

VIEW DESCRIPTION

A View can be described as follows:

- By calling Tables or Views, that is calling segments into another segment;

- By calling Columns via the DB2 VIEW DESCRIPTION (-DBE) screen (see Subchapter ″DB2 VIEW DESCRIPTION″).

COLUMN NAME GENERATION PRIORITIES

There are four different ways of specifying a Column Name. If all four methods are used, the Column Name is generated according to the following order of priority:

1. Column Name entered as an insertion to the system-generated (virtual) lines on the Generation Elements screen of the Block. (See chapter 'Database: Database Block', subchapter 'Generation Elements').
2. Column Name entered on the Call of Elements Screen for a Table or View, 'A*' followed by the Column Name (maximum 8 characters) in the UPDATE TARGET field.

   EXAMPLE:

```
A LIN ELEM.                          UPD/TRGET
  100 COLNE1                         A*COLUMN1
```

1. Column Name entered on a Data Element Description screen with TYPE OF LINE = 'R'.
2. Data Element 6-character code.

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
                         1 2
SEGMENT CALL OF ELEMENTS DZ05 FOURNITURE
3 4     5      7          8 9   10 11  12   13   14        15        16
A LIN : ELEM.  INT.FORM.  U OCC GR K CMD456 CONT VALUE/SFC UPD/TRGET DOC LIBR
  100 : COCARA                     O                                  0067
  110 : NUCOD                      P         S     O                  0067
  120 : FOURNI                                                        0067
* 125 : (-DBE)                                                       0234
  200 : NUCLIE                               S   O                    0067
* 205 : (-DBE)                                                       0234
  210 : DATE                                                          0067
  220 : RELEA                                                         0067
  230 : REFCLI                   V           S   O                    0067
* 235 : (-DBE)                                                       0234
  240 : RUE                      V           S   O                    0067
* 245 : (-DBE)                                                       0234
  250 : COPOS                                S   O                    0067
* 251 : (-DBE)                                                       0234
  255 : VILLE                    V           S   O                    0067
* 256 : (-DBE)                                                       0234
  260 : CORESP                   V           S      O                 0067
      : NAME    : 6

O: C1 CH: -CE
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
SEGMENT CALL OF ELEMENTS DZ05 FOURNITURE

A LIN : ELEM.  INT.FORM.  U OCC GR K CMD456 CONT VALUE/SFC UPD/TRGET DOC LIBR
  270 : REMISE                                                        0067
  280 : MATE                        V                    A*MATERIEL   0067
  310 : PRIX1                                                         0067
  330 : HEURE                                                         0067
  340 : PRECIS                                 S     O                0067
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      : NAME    :
*** END ***
O: C1 CH:
```

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			With the Pactables function, this value is not allowed.
		01-99	Designates a specific Segment. The common part Data Elements are automatically concatenated with each specific part Segment. Although a data element may not be used twice in the same Segment, it may be used in both the common part and in one or more specific Segments (except data structures used as Tables).
3	1		ACTION CODE (REQUIRED)
		'C'	Creation of the line
		'M'	Modification of the line
		'D' or 'A'	Deletion of the line
		'T'	Transfer of the line
		'B'	Beginning of multiple deletion
		'G'	Multiple transfer
		'?'	Request for HELP documentation
		'E' or '-'	Inhibit implicit update
		'X'	Implicit update without upper/lowercase 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

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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
			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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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		KEY INDICATOR FOR ACCESS OR SORT
			For Relational Tables or Views:
		'blank'	Fixed length Column (default value).
		'V'	Variable length Column,

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'W'	For DB2 SQL, SQL/DS, ORACLE, DB2/2 and DB2/6000, generation of a variable length column (VARCHAR).
		'L'	For DB2 SQL, SQL/DS, ORACLE, DB2/2 and DB2/6000, generation of a LONG VARCHAR.
		'C'	For ORACLE V7, generation of a CHAR.
12	6		PRESENCE INDICATOR
			Only the first position of the field is used to specify the nature of a Column presence in a Table.
		'blank'	Optional Column:
			The Column is not required in update and as such will not always be filled in in READ-ACCESS mode. Programming a variable indicator for presence validation is therefore necessary. It is possible to add an optional Column in a Table via the ALTER command.
		O	NOT NULL Column:
			The Column is required in update and is always filled in in READ-ACCESS mode. Programming a variable indicator for presence validation is therefore not necessary. It is not possible to add a NOT NULL Column in a Table via the ALTER command.
			RDMS 1100: This value is not taken into account.
		P	NOT NULL WITH DEFAULT Column (DB2, DB2/2, DB2/6000 and DATACOM/DB), DEFAULT (INTEREL RDBC and RFM), DEFAULT SYSTEM (NONSTOP SQL C30).
			The Column is not required in update but will be initialized to zero or space by the DBMS. As such, it is always filled in in READ-ACCESS mode. It is not possible to add a (NOT NULL WITH) DEFAULT Column via the ALTER command.
			This value is not recognized by RDMS 1100 nor by NONSTOP SQL (Version C10).
13	4		SCHEMA SELECTION
		S	Indicates that the schema has been selected.
			The group and elementary Data Elements must belong to the same sub-schema. This schema must be entered on all the lines.
14	10		SELECTION INDICATOR
		O	Indicates the presence top in a schema.
15	10		UPDATE TARGET

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		A*.....	You may specify a Column Name by entering 'A*' left-justified and followed by the desired name. This name cannot exceed 8 characters in length.
			This value will override the default of the DATA ELEMENT CODE or the value assigned on the Data Element Description screen (if any).
16	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).

Additional View Description (S....DBE)

When Segments are used as Views, an additional screen is available in order to specify where the Columns making up the View come from.

This screen is accessed via the following input in the CHOICE field:

CH: -DBE

The lines entered on the Segment's Call of Elements (-CE) screen are displayed on the Additional View Description (-DBE) screen and cannot be modified there. These lines are identified by an asterisk (*) in the ACTION CODE field.

By inserting lines between the displayed lines, the source of the Column can be specified.

```
EXAMPLE:              A LIN : ELEM.
                        080 : COLUM1
                        100 : COLUM2
                        120 : COLUM3
                        140 : COLUM4
                      O: C1 CH: -CE
```

In order to specify the source of 'COLUM2', you must access the '-DBE' screen, and insert a line via an appropriate LINE NUMBER value, between the Element to describe (100 in this case) and the next element (120).

In the example below, we have selected a LINE NUMBER of 110. Enter the source of the Column in the format of 'ddsseeeeee' where ddss' = the SEGMENT CODE of the Table Segment and 'eeeeee' = the Column's DATA ELEMENT CODE.

```
A LIN : ELEM.                          TAB/ELEM.
* 080 : COLUM1
* 100 : COLUM2
  110 :                                 XX10DATELA
* 120 : COLUM3
* 140 : COLUM4

O: C1 CH: -DBE
```

The source of 'COLUM2' is 'DATELA' of the 'XX10' Table.

NOTE: The identifier expected by the System is the DATA ELEMENT CODE of the Column and not an override like the value (beginning with '*A') entered on the Segment Call of Elements screen.

REMARK - CALLING SEGMENTS

When calling Segments into a View Segment the system automatically tracks the source of the Columns.

EXAMPLE:

View description: (Note: '**' identifies 'XX10' as a Segment.)

```
A LIN : ELEM.              GR
  100 : XX10               **

O: C1 CH: S VV55 CE

Table description:

A LIN : ELEM.
  100 : COLUM1
  120 : COLUM2

O: C1 CH: S XX10 CE
```

Without any Additional View Description (-DBE) specified, the system knows that the COLUM1 Column comes from COLUM1 in the XX10 Table, and that the COLUM2 Column comes from COLUM2 in the XX10 Table.

EXAMPLES OF SEGMENT CALLS

- When a View contains all the Columns of several Tables, instead of calling each Data Element individually into the View description, you can call the

Table segments into the View segment (via the Call of Elements Screen) directly. No additional data is needed on the Additional View Description.

- For example, the View VV10 contains all the Columns of the Table XX10 plus the first Column (COLUM1) from Table XX20.

```
A LIN : ELEM.        GR
  100 : XX10         **
  120 : COLUM1
* 122 : (-DBE)

O: C1 CH: S VV10 CE

A LIN : ELEM.             TAB/ELEM.
* 100 : XX10        **
* 120 : COLUM1
  122 :                    XX20COLUM1

O: C1 CH: S VV10 DBE
```

In this example the View DATA ELEMENT CODE (COLUM1) could be different from the DATA ELEMENT CODE of its source.

USING THE SUB-SCHEMAS

It is possible to select the Data Elements called in the view, by indicating:

- A sub-schema number ranging from 0 to 9, or '*' in the '-DR' screen, in the KEY TYPE field. The '*' means that all Data Elements of the Segment (including the Segment Data Elements called in the Segment) belong to the View.
- In cases of selection by a number ranging from 0 to 9: the user must specify an 'S' (indication of sub-schema) on the Data Elements to be selected in the 'CONT' field of the '-CE' screen and an 'O' in the VALUE/SFC field on the umpteenth Column (n corresponding to the sub-schema number indicated on the '-DR' screen, O being 10).

IMPORTANT NOTE

For reasons of compatibility with batch and on-line generations, it is not possible to specify the sub-schema number on segment calls and group data elements.

In cases of sub-schema utilization, the '-DBE' lines that might be present are ignored.

USING SEGMENT CALLS

- If the KEY TYPE Data Element is blank, it is a View, described by '-DBE' lines. In this case, it is possible to call a Segment that will be considered as a Table. All the Data Elements of the Segment will be selected, without -DBE lines having to be entered.

IMPORTANT: There can be only one level of Segment call.

- If the KEY TYPE Data Element is entered, it is a View described by a selection of sub-schemas or of the entire Table. In case of first level call, this will be considered as a Table. The Segments called in the first Segment will have their Data Elements referenced in this Table, whatever the nesting level.

```
                      ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
ADD'L VIEW DESCRIPTION : DZ05 FOURNITURE
3 4                       1 2                                   5
A LIN : ELEM.                                                  TAB/ELEM. DOC LIBR
* 100 : COCARA                                                              0067
* 110 : NUCOD                                                               0067
* 120 : FOURNI                                                              0067
  125 :                                                        FF10FOURNI    0234
* 200 : NUCLIE                                                              0067
  205 :                                                        CC20NUCLIE    0234
* 210 : DATE                                                                0067
* 220 : RELEA                                                               0067
* 230 : REFCLI                                                              0067
  235 :                                                        CC20RAISOC    0234
* 240 : RUE                                                                 0067
  245 :                                                        CC30RUE       0234
* 250 : COPOS                                                               0067
  251 :                                                        CC30COPOS     0234
* 255 : VILLE                                                               0067
  256 :                                                        CC30VILLE     0234
* 260 : CORESP                                                              0067
      : NAME    :

O: C1 CH: S dz05 DBE
```

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			With the Pactables function, this value is not allowed.
		01-99	Designates a specific Segment. The common part Data Elements are automatically concatenated with each specific part Segment. Although a data element may not be used twice in the same Segment, it may be used in both the common part and in one or more specific Segments (except data structures used as Tables).
3	1		ACTION CODE (REQUIRED)
		'C'	Creation of the line
		'M'	Modification of the line
		'D' or 'A'	Deletion of the line
		'T'	Transfer of the line
		'B'	Beginning of multiple deletion
		'G'	Multiple transfer
		'?'	Request for HELP documentation
		'E' or '-'	Inhibit implicit update
		'X'	Implicit update without upper/lowercase 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	10		COLUMN SOURCE
			Enter the SEGMENT CODE of the Table segment and the DATA ELEMENT CODE to specify the source of the Column.
			Use the format: ddsseeeeee.
			In batch mode, this corresponds to the UPDATE TARGET field (columns 71 to 80).

Chapter 5. SQL Accesses

Implementation Made Easy

With the Pacbench C/S, O.L.S.D. and Batch Functions, you can implement SQL accesses easily (connect, disconnect, commit...), without having to code them entirely in procedural code. The generated code automatically adapts to the Block type.

CODING RULES

In Procedural Code lines (-P), you must enter one of the following operators in the OPERATOR field and the Block code in the OPERANDS field.

This is the list of accesses you can implement and the syntax you must follow:

```
CONNECT (or equivalent)     : SCC cccccc d
DISCONNECT (or equivalent)  : SDC cccccc d r
COMMIT                      : SCO cccccc d
ROLLBACK                    : SRO cccccc d
WHENEVER                    : SWH statement
cccccc: VA Pac code of the Block (6 characters)
d: value '2' if distributed database (e.g. Oracle, Sybase)
r: value 'R' for DISCONNECT with ROLLBACK.
The d and r indicators can be reversed.
```

Each statement can be written on more than one line (without operator on continuation lines). On these continuation lines, you can enter options available with some RDBMS (e.g. FORCE option in a COMMIT statement for Oracle).

Ex.: Coding a CONNECT statement to the Oracle BLOCOD Block:

```
 OPE  OPERAND
 SCC  BLOCOD  2
```

GENERATION RULES

In the Segment Calls (-CS for Pacbench/CS and OLSD, or -CD for the Batch Function), if you declare an SQL organization (ORGANIZATION field) and so a Block code (EXTERNAL NAME field), this organization has priority on the Block type indicated on the Bloc Definition.

In the Segment Calls, if the Block is indicated as being distributed, the accesses related to this Block will be generated as 'distributed'.

Unknown SQL statements are ignored.

END-EXEC is automatically generated and with the batch generator, it is always followed by a period.

Customized SQL Accesses

Introduction

WARNING: You must be familiar with the SQL syntax before customizing SQL accesses in VA Pac.

INTRODUCTION

You can:

- add or replace clauses in the standardly generated access. To do so, you can modify the standard access or associate the standard access with a non-standard one.
- create a new SQL access by coding a non-standard access,
- implement extraction criteria which correspond to extraction methods.

The implementation of customized standard accesses is automatic or almost automatic whereas the implementation of non-standard accesses must be requested in procedural code.

So you choose between customizing a standard access or a non-standard one according to how often this access will be used in Screens or Programs.

With VA Pac, you write customized SQL accesses on the Generation Elements lines of a Segment (CH: S....GG) and implement them or ignore them in the entities which use them. As a matter of fact you can choose to implement them in a given Screen but not in another one, or to implement only some of them... You are quite free to use them as you need.

A customized access is, for example, to select only the clients who put in an order which amounts to more than $50.

Description

You describe customized accesses on the Segment Generation Elements screen (S....GG).

On the first line, you enter the access you want to customize and on the following lines, you specify the part(s) of the access you want to modify or add.

The storage capacity of SQL accesses is no more limited to 1000 lines on the -GG screen of Segments. It is possible to create more lines as the field 'NLG' (line number) is alphanumeric now.

FIRST LINE

Enter value 'G' in the LINE TYPE field.

In C2 option, the DESCRIPTION field contains 2 parts (tab from one to the next):

- in the first part, enter 'SQL' to indicate that you want to customize an SQL access.
- in the second part, enter the access type you want to customize. The access type is standard or non-standard.

STANDARD ACCESSES

The standard access types, which will be automatically implemented, are:

```
R  : Select
RU : Select for Update
RA : Select (Pacbench C/S)
W  : Insert
RW : Update
D  : Delete
P  : Open Cursor
RN : Fetch
DC : Declare Cursor
CL : Close Cursor
UC : Update Cursor (Pacbench C/S)
```

NOTE: DC (Declare Cursor) standard access is not managed for SQL accesses where DC is generated in the Working Storage Section.

```
Example of the customization of the standard select access:
 T  DESCRIPTION
 G  SQL   R
```

NON-STANDARD ACCESSES

A non-standard access must be coded on any one or two characters. This access can be:

- combined with a standard access if you want to modify only part of the standard access and leave the generator to manage the syntax. In this case you must enter the code of the standard access first and, after a blank, the code of the non-standard access.
- or used alone if you want to create an entirely new access. In this case you must describe this access entirely and implement it in procedural code. You must manage the syntax (punctuation, keywork...) entirely.

Example of the customization of a non-standard select access:

```
T  DESCRIPTION
G  SQL   R1 RA
```

Example of the creation of a non-standard access which counts the number of records in the clients table.

```
T  DESCRIPTION
G  SQL   CP
```

EXTRACTION METHODS

An extraction method is a select access to a Business Component. So it is specific to Pacbench C/S.

Four access statements are required: Declare Cusor, Open, Fetch, Close.

You can then:

- partially describe only one customized access, by entering the 'EX' access type twice.
- or describe the four accesses entirely, by combining your customized accesses with each of the four accesses (DC, P, RN, CL).

Example of the declaration of the METHODNAME extraction method:

```
T  DESCRIPTION
G  SQL   EX EX METHODNAME
```

Example of the customization of the METHODNAME four accesses:

```
T  DESCRIPTION
G  SQL   D1 DC METHODNAME
G  SQL   O1 P  METHODNAME
G  SQL   F1 RN METHODNAME
G  SQL   C1 CL METHODNAME
```

You must enter the method name in uppercase letters, from column 7, on 10 characters maximum.

FOLLOWING LINES

An SQL access is described by clauses (access statement, where, order...). Each clause is described by a keyword and data related to the keyword. From the second line onwards, you enter these keywords, which apply to the customization of both standard and non-standard accesses.

In the LINE TYPE field, enter value 'G'.

In the first part of the DESCRIPTION field, enter the keyword and in the second part, enter the parameters required for its operation.

The corresponding data will replace the data which would have been generated otherwise. If it must be added as a complement, enter 'ADD' before or after each keyword.

A keyword described on more than one line should not be repeated on each line.

```
KEYWORD    PARAMETERS
ACCESS     SQL Access
FROM       List of tables
ORDER      List of columns
COLUMN     List of columns and host-variables
WHERE      Selection criteria
PARAM      Additional parameters
```

See next paragraphs for a detailed explanation on each keyword.

The use of these keywords is illustrated in examples at the end of this subchapter and at the end of the chapter.

The generated format of date and variable Data Elements (e.g. CURDAT (current date) column entered as a parameter of the ORDER statement) varies according to the Block type.

PUNCTUARION RULES

The corresponding delimiter is automatically generated at the end of each line. If a column name and/or a host-variable name exceeds one line, indicate this line as being 'V'-type and continue on the next line ('G'-type line).

In this case, and if you enter a column name AND a host-variable name (with or without : -), you must code the host-variable on a new line.

On the other hand, if you code a host-variable AND its indicator under the form : - (without Segment code), you must code them on the same line.

The parameters associated with the COLUMN keyword have additional punctuation rules, which are detailed in the 'COLUMN' paragraph below.

1. ACCESS

Allows you to rewrite the clause of an access (SELECT, UPDATE, DELETE ...).

2. COLUMN

This keyword is followed by the list of columns.

These columns can be coded either in the SQL-Data Element form (in this case they are replaced during generation by their relational name) or directly under their relational name.

Each column can be followed, after one blank at least, by the code of the host variable (and indicator), coded either in the form ':-Data Element code' (during generation, the Segment code defined on the -CS screen (Screen Segment code) will be inserted before the :-Data Element code), or under the name of the host variable.

If the host variable is not coded in an access where it is required, it will be generated with the Screen Segment code and the Data Element indicated in the column (in the SQL-Data Element code form, if not the code of the host variable must be indicated).

Specific punctuation rules:

If the column and host-variable names exceed one line, you can use one or more 'V'-type lines but only for the host-variable name; the column name is entered on one line only.

For the UPDATE statement ('RW') of the COLUMN keyword, the host-variable and its indicator must be coded on the same line whatever their input format.

3. FROM

This keyword is followed by the list of tables. At generation time, FROM is either transformed into INTO or remains FROM according to the access type.

4. WHERE

Allows you to indicate the selection criteria. The column names can be coded in the 'SQL-Data Element code' form and the host variables and indicators in the ':-Data Element code' form.

5. ORDER

Allows you to indicate the Order By.

Same principle as WHERE for coding the column names.

6. PARAM

Allows you to code additional parameters such as COUNT, HAVING, ... The ADD keyword is not necessary.

Same principle as WHERE for coding the column names.

EXAMPLES:

Customization of a non-standard select access to select a client number higher than or equal to the requested number. You modify the WHERE clause only but you do not want to modify the standard access directly because this selection applies to few Screens only. So you enter:

```
T  DESCRIPTION
G  SQL     RB RA
G  WHERE   SQL-NOTJ >= :-NOTJ
```

Creation of a non-standard access to count the number of records in the clients table. In this case it is really a specific need. So you describe a new access in the following way:

```
T  DESCRIPTION
G  SQL     CP
G  ACCESS  SELECT COUNT (*)
G          INTO :WWCA-COUNTER
G          FROM PTB0.CLIENT
```

Generating keywords is possible only for standard or similar accesses (CP R). In the case of a CP access (which is not followed by a standard access), you must also write the keyword on the 'G' line.

EXTRACTION METHODS

If an extraction method is specified in the selection request, the selection of the occurrence list is performed according to one or more criteria.

You can:

- describe only one 'EX' customized access with the required clauses:

```
T  DESCRIPTION
G  SQL    EX EX METHODNAME
G  WHERE  ...
G  ORDER  ...
```

- or entirely describe the four customized accesses, and specify the clauses for each one:

```
T  DESCRIPTION
G  SQL    D1 DC METHODNAME
description...
G  SQL    O1 P  METHODNAME
description...
G  SQL    F1 RN METHODNAME
description...
G  SQL    C1 CL METHODNAME
description...
```

See also the information included in the Developer's documentation series, eBusiness Applications, 'Pacbench C/S Application & Business Logic', chapter 'Business Component', subchapter 'Inserting Specific Code', 'Insertion Relative to the 'Physical Accesses' Level', Customized SQL Accesses' part.

OPTION

The option UPPER=NO, entered on an O-type line, inhibits the lowercase/uppercase conversion of values entered between two literal delimiters (the delimiter is defined at the Library level).

RESTRICTIONS

The syntax of SQL statements is not implemented on RDMS blocks ('R' organization).

The prefixing rule is not applied. The table name is kept as it is and the period located at the beginning is deleted if it exists.

WARNING: In the case of program-macro and macro-macro overlaps, the lines generated from the Generation Elements screen (CH: S....GG) cannot be deleted.

Implementation

PACBENCH C/S AND O.L.S.D. FUNCTIONS

O.L.S.D. Prerequisite: customized SQL accesses are taken into account if the PACSQL option has been entered on the -O Dialogue Complement.

All accesses are generated in function 80.

- Customized standard accesses

  Customized standard accesses are automatically generated with the customization entered on the Segment's Generation Elements lines (S....GG).

  On the other hand, if you want to ignore the customization of standard accesses, you must specify it in the O.....P lines.

- Non-standard accesses

  To implement non-standard accesses, you must specifically call them in the O.....P lines.

NOTE: Standard accesses are ignored for R (SQL RDMS) and 2 (DB2 without generated access) organizations but you can request the implementation of non-standard accesses.

CODING IN -P LINES

You code the implementation (or non-implementation) of customized accesses in a sub-function dependent of the F80 function.

You must enter *C in the 'Condition type' column and the Segment code in the 'Execution condition' column.

- To ignore the customization of a standard access, you must enter the standard access code after the Segment code.
- To implement a non-standard access, you have three possibilities:
  - If you enter the standard access code before the non-standard access code, the customization of the non-standard access will be generated in the standard processing of the F80 function.
  - If you enter the non-standard access code alone (without specifiying any standard access), the customization of the non-standard access will be generated in a specific sub-function of function F80. It will then constitute a user function unknown to the generator, so you will have to describe and manage it in procedural code entirely.
  - If you enter the non-standard access code followed by the '=' sign and the standard access code, the customization of the non-standard access will be completed by the processing automatically generated for the specified standard access in the F80 function.

    Examples of coding:

```
LVTY CONDITION
10*C SS00 R     the customized standard select access
                entered on S SS00 GG will be ignored
```

```
10*C SS00 R R1  the R1 non-standard select access entered
                on S SS00 GG will be implemented and will
                complete the automatically generated read
                processing of SS00 Segment.
10*C SS00 R1    the R1 non-standard select access entered
                on S SS00 GG will be implemented and
                generated in a specific function
                F80-SS00-R1.
10*C SS00 R1=R  the R1 non-standard select access entered
                on S SS00 GG will be implemented and
                completed by the processing automatically
                generated for the standard select access.
```

- Extraction Methods

If you described an 'EX' customized access, you must call this access explicitly. So you create a sub-function which depends on function 80 and enter *C in the 'Condition type' and the Segment code in the 'Execution condition' field. You must also enter EX and the method name in uppercase letters.

Examples of the implementation of the METHODNAME extraction method:

```
LVTY  CONDITION
10*C  SS00 EX METHODNAME
```

If you described four customized accesses (DC, P, RN, CL), you must implement these four accesses in function 80 (same coding as with 'EX' access, except that 'EX' is replaced by 'D1', 'O1', 'R1', 'C1').

Implementation of the four customized accesses in -P:

```
LVTY  CONDITION
10*C  SS00 D1 METHODNAME
10*C  SS00 O1 METHODNAME
10*C  SS00 F1 METHODNAME
10*C  SS00 C1 METHODNAME
```

For more details, refer to the 'Pacbench C/S Application & Business Logic' Manual, chapter 'Business Component'.

BATCH FUNCTION

In the Batch Function, customized accesses can be implemented only if the Program -CD contains a Block code in the EXTERNAL NAME field and an organization in the ORGANIZATION field.

You must explicitly call each customized access you want to generate in the Program procedural code lines (P.....P). In the OPERATOR field, enter 'SQL' and in the OPERANDS field, enter:

- the Segment code in the Program
- the Segment code in the Library if it is different from the Segment code in the Program,
- the access type:
  - If you specify the standard access before the non-standard access, the customization of the non-standard access will complement the standard access.
  - On the other hand if you specify only the non-standard access without specifying a standard access, the customization of the non-standard access will be taken into account. It will constitute a user function unknown to the generator. You will have to describe and manage it in procedural code entirely.

The customization of a standard select access from Segment SS00 (Segment code in Program: PGSG) will be specified in the following way:

```
OPE  OPERANDS
SQL  PGSG SS00 R
```

The customization of a non-standard select access from Segment SS00 (Segment code in Program: PGSG) will be specified in the following way:

```
OPE OPERANDS
SQL PGSG SS00 R R1
```

Examples

EXAMPLE OF CUSTOMIZED ACCESSES

In the following example, two standard accesses have been customized: the select access and the update access.

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDMB.DDDD.BMS.241
GENERATION ELEMENTS FOR SEGMENT    PI00

A LIN : T DESCRIPTION                                                   LIBR.
  100 : G SQL                   R1 R                                     *DCC
  110 : G WHERE ADD             AND SQL-PILNAM LIKE W-PILNAM             *DCC
  120 : G SQL                   W1 RW                                    *DCC
  125 : G ACCESS                UPDATE                                   *DCC
  200 : G COLUMN                SQL-QUSAM                                *DCC
  205 : G                       SQL-DAQUA                                *DCC
  210 : G WHERE ADD             SQL-COQUA > 'A'                          *DCC
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: SPI00 GG
```

EXAMPLE OF AN EXTRACTION METHOD

The following example shows extraction methods, named 'CUSTNAME' and 'TOWN'.

The first screen shows their writing in the CN10 Segment's Generation Elements lines.

The second screen shows their implementations in the CLCNT Screen Procedural Code lines.

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDMB.DDDD.BMS.241
GENERATION ELEMENTS FOR SEGMENT    CN10

A LIN : T DESCRIPTION                                                    LIBR.
  100 : G SQL                   EX EX CUSTNAME                          *DCC
  110 : G WHERE                 CUSTOMER_NUMBER >= :7-WW00-CUSTNB       *DCC
  120 : G                       AND CUSTNA >= :7-WW00-CUSTNA            *DCC
  125 : G                       ORDER BY CUSTOMER_NUMBER                *DCC
  200 : G SQL                   EX EX TOWN                              *DCC
  205 : G WHERE                 TOWN = 'PARIS'                          *DCC
  210 : G                       ORDER BY CUSTOMER_ NUMBER               *DCC
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: SCN10 GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDMB.DDDD.BMS.241
PROCEDURAL CODE       O CLCLNT  Customers Server              FUNCTION: 80

A SF LIN OPE OPERANDS                          LVTY CONDITION
  DD     N   PHYSICAL ACCESS FOR CUSTNAME     10*C CN10 EX CUSTNAME
- -- --- --- -------------------------------- ---- -------------------
  EE     N   PHYSICAL ACCESS FOR TOWN         10*C CN10 EX NAME
- -- --- --- -------------------------------- ---- -------------------









O: C1 CH: OCLCNT P
```

Chapter 6. Database Blocks

Database Definition (B.....)

A Database is defined through a Database Block, accessed via the following input in the CHOICE field:

CH: B.....

The block is defined with a code, a name and a specific TYPE:

Q2: for DB2 blocks,

Q3: for SQL SERVER blocks,

QB: for DB2/2 and DB2/6000 blocks,

QC: for DATACOM/DB blocks,

QN: for NONSTOP SQL blocks,

QO: for ORACLE blocks, releases earlier than V6,

QP: for ORACLE blocks, from release V6 on,

QR: for RDMS 1100 blocks,

QS: for SQL/DS blocks,

QT: for INTEREL RDBC blocks,

QU: for INTEREL RFM blocks,

QY: for SYBASE blocks.

The external name of the database is normally specified via an input in the DATABASE BLOCK EXTERNAL NAME field.

The EXTERNAL NAME OF THE SCHEMA field is not used for TYPE OF BLOCK 'Qx'. Therefore once 'Qx' is entered, the field will no longer appear on the screen.

ASSOCIATED LINES

Generation Elements (-GG).

The physical information necessary to generate the database is entered on the Generation Elements lines associated with the Block, in order to complement the logical information entered on the Database Block Definition.

Generation Options (-GO)

In this screen, you specify options such as the prefixing mode, the generation of COMMIT...

Comments (-GC)

In this screen, you enter comments on the Database Block or on the objects it calls.

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
                                 1
BLOCK DEFINITION.............:    Q2BLOC

NAME..................:2 DB2 EXAMPLE
TYPE..................:3 Q2 D.B.2 SQL
VERSION...............:4


EXTERNAL NAME.........: EXTQ2DB2   5



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


EXPLICIT KEYWORDS..: 8


SESSION NUMBER.....: 0067        LIBRARY......: BMS    LOCK....:




O: C1 CH: B q2bloc                     ACTION:
```

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'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).
			RELATIONAL DATABASES
		Q2	DB2 SQL
		Q3	SQL SERVER
		QB	DB2/2 and DB2/6000
		QC	DATACOM/DB
		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
		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,

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'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
			AS/400 PHYSICAL FILE
		PF	AS/400 Physical file (IBM SYS. 38)
		LF	AS/400 Logical file (IBM SYS. 38).
			DMSII DATABASE
		20	DMSII Database (DASDL)
4	4		VERSION NUMBER
			Version number of the database system.
		3000	NONSTOP SQL: Version C30
		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.

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

Database Description (B.....DR)

A relational database is described on a Database Block Description screen accessed via the following input in the CHOICE field:

CH: B.....DR

The description is the list of Objects which make up the database.

Seven different SQL RECORD TYPES are available on this screen. They appear in the logical order of their creation.

```
.SPACE          ------> P
.TABLE          ------> T
.VIEW           ------> V
.INDEX          ------> I
.ALTER TABLE    ------> A
.PRIMARY KEY    ------> K
.FOREIGN KEY    ------> J
```

BLOCKS / OBJECTS	P	T	V	I	A	K	J
Q2	Y	Y	Y	Y	Y	Y	Y
Q3	-	Y	Y	Y	Y	Y	Y
QC	Y	Y	Y	Y	Y	Y	Y
QN	-	Y	Y	Y	Y	-	-
QO	Y	Y	Y	Y	Y	-	-
QP	Y	Y	Y	Y	Y	Y	Y
QR	Y	Y	Y	Y	Y	-	-
QB	-	Y	Y	Y	Y	-	-
QS	Y	Y	Y	Y	Y	Y	Y
QT	-	Y	Y	Y	Y	-	-
QU	-	Y	Y	Y	Y a	-	-
QY	-	Y	Y	Y	Y	Y	Y
DB	Y	Y	Y	Y	Y	Y	Y

NOTE:

1. column add-on only
1. Objects are related to the Space which precedes them. An 'Alter Table' must follow the Table it modifies. An Index must follow the Table to which it is related.

It is not possible to create an Index associated with a View; the generated 'CREATE INDEX' command will be rejected when the catalog is updated.

For RDMS, the Primary Key is not generated by an I line; it is included in the generation of the Table that precedes it (T line).

The type of Index, Primary key or Foreign key may not be modified: these lines must be deleted, then created with another type.

The table code entered on Index, Primary key or Alter lines may not be modified.

Some objects are specifically taken into account in Oracle V7, SYBASE and SQL SERVER:

For Oracle V7: Package, Function, Procedure and Trigger.

```
.PACKAGE  ------> C   (function and procedure package;
                      The difference between the Package
                      BODY and its definition is made
                      clear by a 'B' in the key type)
.FUNCTION ------> E
.PROCEDURE------> Q
.TRIGGER  ------> R   for ORACLE V7, SYBASE, SQL SERVER
```

GENERATION ELEMENTS

The Database Description (-DR) screen provides the System with the logical information needed to generate DDL statements. The SQL RECORD TYPE causes system-generated 'virtual' lines to be incorporated; these lines are displayed and updated on Generation Elements (-GG) lines related to Database Block Description lines.

NOTE: On I-type lines, a + sign indicates that keys have been specified for this call (-K). If an * character appears at the end of the line, it means that -GG lines are associated to the Object called (-DRnnnGG).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
RELATIONAL BLOCK DESCRIP. 1 Q2BLOC DB2 EXAMPLE
2 3     4  5                            6              7    8
A LIN : T EXTERNAL NAME                TABLE CODE     KEY  GEN           LIBR.
      :                                VIEW           TY   CDE
  080 : P ESP1-TABLE-Q2BLOC                                C              0067
  100 : T DODZ05                         DZ05              C              0067
  110 : K                                DZ05              C * +          0048
  130 : V VUDZ05S3                       DZ05         3    C              0058
  200 : T DODZ10                         DZ10              C *            0067
  210 : I INDZ10                         DZ10              C   +          0048
  220 : J CEXISTF  DZ05                  DZ10         C    C              0048
  300 : T                                FO10              C              0067
  350 : V VUDZ09DBE                      DZ09              C              0067
  360 : V VUDZ09S4                       DZ09         4    C              0048
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: B q2bloc DR
```

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
4	1		SQL record type
		'P'	Tablespace (except for Interel RDBC, Interel RFM, Nonstop SQL, Sybase and SQL Server)
		'T'	Table
		'V'	View
		'I'	Index
		'A'	Alter Table: Column updating
		'K'	RDMS 1100: Primary Key (Processed with the generation of the table that precedes it.)
			DB2, Datacom/DB, SQL/DS, Oracle V6 and V7, DB2/2, DB2/6000, Sybase and SQL Server: Primary key (Processed with the generation through an ALTER TABLE command.)
		'J'	DB2, Datacom/DB, SQL/DS, Oracle V6 and V7, Sybase and SQL Server: Foreign key (Processed with the generation through an ALTER TABLE command.)
		'C'	Package (Oracle V7 only)
		'E'	Function (Oracle V7 only)
		'Q'	Procedure (Oracle V7, Sybase, SQL Server)
		'R'	Oracle V7, Sybase and SQL Server: Trigger
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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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.
		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	Modification This choice is possible on Table-type lines only, on all Database types, except SQL/400. It generates an ALTER statement.
		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'.

Input of Key or 'Alter Table' (-DRnnnK)

The Input of Data Element of Key screen is used to select the Columns making up the Table key or the Columns selected in an Alter Table.

It is accessed via the following CHOICE:

B.....DRnnnK

where 'nnn' is the line number of a Block Description line whose type is I, K, A or J.

This screen may also be obtained by positioning the cursor on the desired description line and pressing the relevant PFkey (standard PF9).

INDEX COLUMNS (I):

The Input of Data Element of Key screen displays the list of Data Elements called into the Segment specified on the corresponding I- or K-Type Block Description line.

The Columns making up the key are selected by entering a KEY RANK NUMBER from 1 to 16; the numbering sequence cannot contain gaps and must begin with 1, which corresponds to the major sort criterion.

To erase the value of this field, the user enters a non-blank alphabetic character.

The SORT ORDER field indicates the sort order: the possible values are A for ascending order, D for descending order, and a blank (no sort DDL command will be generated and the default DBMS command will be used).

A window on the right side of the screen displays the configuration of the key.

ALTER TABLE COLUMNS (A)

The Input of Data Element of Key screen displays the list of Data Elements called into the Segment specified on the corresponding A-Type Block Description line.

The Columns are selected by entering a RANK NUMBER from 1 to 16; the numbering sequence cannot contain gaps and must begin with 1, which corresponds to the major sort criterion.

To erase the value of this field, the user enters a non-blank alphabetic character.

The ORDER field indicates the modification type : values are Blank for column adding (default value), D or A deletion and M for modification.

DATABASES	blank	M	A or D
DB2 - DB2/6000	X		
ORACLE V7	X	X	
DATACOM	*	*	*
NONSTOP	X		
RDMS	*	*	*
SQL/DS	X		
INTEREL RDBC	X		X
INTEREL RFM	X		

'*' indicates that the three modification types can be used simultaneously.

'X' indicates that the three modification types must be used separately.

NOTE:

COMPOSITION OF PRIMARY KEYS AND FOREIGN KEYS (K, J)

The Data Elements of the called Segment are displayed: you can select Data Elements by entering a generation rank number in front of them.

The rank number is comprised between 1 and 16 and must not include any gap. The ORDER field is ignored.

NOTE: If a Data Element used as a key is called in a Segment, it must be cancelled as a key before being cancelled in the Segment.

```
 --------------------------------------------------------------------------------
|                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260|
| INPUT OF DATA ELEMENTS 1    QCBLOC DATACOM/DB EXAMPLE                   700 |
|                             DZ10                                            |
|                               2    3             --------------------        |
|                               RANK ORDER        |      COLUMN       |   LIBR.|
|  COCARA                                        |   QTMLI      A    |   0067 |
|  NUCOM                        03               |   FOURNP     M    |   0067 |
|  FOURNP                       02    M          |   NUCOM           |   0067 |
|  QTMLI                        01    A          |   QTMCO      D    |   0067 |
|  QTMCO                        04    D          |   INFOR           |   0067 |
|  INFOR                        05               |                   |   0067 |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
|                                                |                   |        |
| *** END ***                                                                 |
| O: C1 CH: -DR210K                                                           |
 --------------------------------------------------------------------------------
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		BLOCK CODE
			One to six alphanumeric characters.
2	2		RANK
			KEY DEFINITION --------------
		1-16	This field is used to indicate the rank of the Elements making up the Table key (Primary , Foreign Keys or Index). The rank is selected by entering a number from ’1’ to ’16’ in this field.
			The selection must begin with ’1’, which is the major sort criterion, and cannot include gaps.
			ALTER TABLE ----------- It is also used to indicate columns taken into account for the modification of a Table, by indicating the generation order of the ALTER requests. The Elements are selected by entering a number from ’1’ to ’16’ in this field.
			The selection must begin with ’1’, and cannot include gaps.
			In order to reset this field to blank, enter a non- numeric character.
3	1		SORT ORDER OR MODIFICATION TYPE

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			KEY DEFINITION ----------
		'A' 'D' Blank	The Column is sorted in ascending order. The Column is sorted in descending order. Default value: no DDL sort command is generated.
			ALTER TABLE ----------
		Blank	Default value : the Column is to be added.
		A or D	The Column is to be deleted.
		M	The Column is to be modified.

Generation Elements and Options

The Database Block Definition screen provides the System with the logical information necessary for source language generation.

The physical information is indicated on the Generation Elements (-GG) lines of the Block.

The Database Block Description (-DR) screen provides the logical information necessary for the generation of the DDL of the Spaces, Tables, Views and Indexes.

The associated physical information is entered on Generation Elements lines associated with each description line (-DRnnnGG).

Virtual lines and P.I.A.'s facilitate entry of the physical information.

Generation Elements lines to be taken into account during generation must have the value 'G' in the TYPE OF LINE field.

NOTE: Generation Elements lines are only taken into account for the generation of 'CREATE' and 'ALTER TABLE' commands generated through the '-DR' screen from 'A' type lines.

MANAGEMENT OF OPTIONS

To specify options, create an O-type line in the Generation Options screen of a Database Block (B.....GO). On this line, you can indicate the following options:

- Default punctuation

  With this option (COMMA=YES), the punctuation will be automatically generated at the end of the lines on which you have added or modified the characteristics of of the Table/View called in the Block.

- No default punctuation

  This is the default option of all the SQL Blocks, except DB2 SQL, SQL/DS and ORACLE V5.

  With this option (COMMA=NO), you must enter the punctuation when you add or modify the characteristics of columns in the 'Generation Elements' (-GG) of the Table/View called in the Block.

- No generation of VA Pac Constants in Comments

  Enter this option (COMMENT=NO) to prevent generating the VA Pac Constants in comments when generating the Block.

- Generation of COMMIT statements

  With this option (COMMIT=YES), a COMMIT statement is generated each time an SQL object is cancelled ('DROP'). In the same stream, you can then cancel and re-create the same object.

- No standard prefixing

  If you enter this option (PREFIX=NO), the standard prefixing of the Tables' and Views' external names will not apply in this Block. For information on the standard prefixing, refer to Chapter 'Table or View: Segment', Subchapter 'Table or View Definition', paragraphs related to the prefixing.

  With this option, an external name which does not include a dot or a prefix will be generated as is, and an external name entered as '.NAME' will be generated with a prefix which is the user's code.

- Return code set to '06'

  With this option (RETCOD=06), the return code is set to '06' instead of '08' when errors are detected upon the Block generation. The generation sequence does not stop and goes to the next step.

- Generation of Date, Time and Timestamp fields (NonStop Tandem)

  With this opion (TIME=YES), Date (with the 'M' format), Time ('T' format) and Timestamp ('TS' format) fields will be recognized in NonStop Tandem Blocks whose version number has been left to blank ('Other Systems, all versions').

- 'No generation of Date, Time and Timestamp fields NonStop Tandem)'

  With this opion (TIME=NO), Date (with the 'MM/DD/YYYY' format), Time and Timestamp fields will be rejected in NonStop Blocks whose version number has been left to blank ('Other Systems, all versions').

- No conversion of lowercase into uppercase

  With this option (UPPER=NO), lowercase letters will not be converted into uppercase letters upon generation.

- Complete description of a DB2 Table

  With this option (DESCR=ALL), the complete description of the Segments which constitute DB2 Tables will be generated.

All the fields called in the Segments (i.e. (i.e. their group fields, redefines and occurs will be described in the Working Storage Section. occurs) will be generated in the Working Storage Section.

This option is also required to break down dates into elementary fields for DB2 Segments (the BREAKDATE=YES option must be indicated on the Library, Program, Dialogue or eBusiness Application Generation Options (-GO)).

GENERATED LINES

The System automatically displays lines on the Generation Elements screens, in order to guide you when you enter the Block's physical information. These lines are identified by an asterisk in the ACTION CODE field and '*VIRT' or '*GUID' in the LIB field. Collectively, these lines are called 'virtual' lines.

The lines identified by '*VIRT' are generated.

The lines identified by '*GUID' must be created to be taken into account at generation time. To do this, the '*' must be deleted if input is entered on the line, or if no input is necessary, an explicit ACTION CODE ('C' or 'X') must be used.

The physical characteristics of a Database and the elements which compose it are entered on the Generation Elements (-GG) screen via Parameterized Input Aids, Virtual lines, and user modifications, as needed.

You may create new lines or modify the 'virtual' lines generated by the DBD function.

Virtual lines are identified by a generated LINE NUMBER:

To create additional lines, insert a line with an appropriate LINE NUMBER at the desired place.

To modify an existing virtual line, repeat the LINE NUMBER of that line and enter a modification of the line.

A 'G' entered in the TYPE OF LINE field means that the lines are to be taken into account at generation time.

EXAMPLE: Insertion of lines

```
A LIN : T DESCRIPTION
* 100 : G CREATE TABLE (EXTERNAL NAME) IN DATABASE-NAME
  160 : G  These two lines will be inserted right after
```

```
  170 : G  the CREATE statement
* 200 : G       ---> COLUMN INSERTION BEGINNING <---
* 300 : G       ---> COLUMN INSERTION END <---
```

A semi-colon (';') is generated before the inserted line if it begins in position 1 of the COMMENT field.

OVERRIDING COLUMNS

INSERTIONS:

The generated 'Virtual' lines indicate where to place the insertion lines via "--> COLUMN INSERTION BEGINNING <--" and "--> COLUMN INSERTION END <--". Use LINE NUMBERs that fall between the LINE NUMBER values assigned to these lines.

To override the Column's pre-defined values, you identify the Column by its DATA ELEMENT CODE left-justified between a 'less than' sign (<) and a 'greater than' sign (>). The two delimiters must be six positions apart (example: <LIBFO >). values. After identifying the Column, you enter the desired values, beginning in column 2.

To complete the generated data, you enter, after <LIBFO >, a space and the 'ADD' keyword, followed by the desired data. Enter the end mark on the column declaration line because it will not be automatically generated.

EXAMPLE: Modification of a Column format:

```
A LIN : T DESCRIPTION
* 100 : G CREATE TABLE (EXTERNAL NAME) IN DATABASE-NAME
* 200 : G       ---> COLUMN INSERTION BEGINNING <---
  210 : G <LIBFO >
  215 : G     DO10LIBFO   FLOAT
* 300 : G       ---> COLUMN INSERTION END <---
```

In this case, the name will be "DO10LIBFO", the format will be "FLOAT".

NOTES:

- This method cannot be used to delete a Column.
- For DB2, the comma (separator) is generated at the end of each line. In order to avoid this generation, a '&' must be inserted after the last non-blank character of the line.

'SOURCE' COLUMNS

INSERTIONS:

To insert a Column into a Table, the same technique as above is used : the start and end of the insertion area is indicated above: the start and end of the insertion area is indicated on the Virtual lines. LINE NUMBERs that fall between the start and end LINE NUMBERs are appropriate for this data.

In order to override a source Column of a Column in a View the user must create a 'G'-type line formatted as follows <FFNN-DELCO >, FFNNDELCO being the code entered in the SOURCE COLUMN field (labelled TAB/ELEM.) on the '-DBE' screen.

EXAMPLE:

```
A LIN : T DESCRIPTION
* 100 : G CREATE VIEW (VIEW NAME)
* 200 : G     ---> COLUMN INSERTION BEGINNING <---
* 300 : G     ---> COLUMN INSERTION END <---
* 400 : G  AS SELECT ALL
* 500 : G     ---> SOURCE COLUMN INSERTION BEGINNING <---
  550 : G  <FFNN-DELCO >
  560 : G   'X'
* 600 : G     ---> SOURCE COLUMN INSERTION END <---
* 700 : G     FROM (SOURCE TABLES NAME)
```

You must manage the Source Table (after the FROM clause).

INSERTION OF 'LABELS' AND 'COMMENTS'

At the time of generation, the labels and comments of the Tables and Columns are, by default, the codes and names entered on the definition screens. However you can replace them by labels and comments you enter on the comments lines of the Table or the View.

- Insertion of Labels

  You can enter labels only in the DB2, SQL/DS database blocks, on the -GC screen of lines T and V.

  The maximum length of labels is 30 characters. You must leave the LINE TYPE blank and enter the '+LAB' command in column 1, followed by the label in the case of a Table or View, or followed by <DELCO > and the label in the case of a Column.

  If you have modified the same object several times on the -GC screen, only the last modification line will be taken into account.

- Insertion of Comments

  You can enter comments only in the DB2, SQL/SD, ORACLE and NONSTOP/SQL database blocks, in the -GC screen of lines T, V and I (only for NONSTOP/SQL).

  You must leave the LINE TYPE blank and enter the '+COM' command in column 1, followed by the comments in the case of a Table, View or Index, or by <DELCO > and the comments in the case of a Column.

You can enter the comments on several lines. At the time of generation, a maximum length of 254 characters, which varies according to the SQL in use, will be taken into account.

EXAMPLE: Insertion of labels and comments:

```
A LIN : T DESCRIPTION
  900 :   +LAB NEW LABEL OF THE TABLE
  910 :   +LAB <DELCO > NEW LABEL OF THE COLUMN
  920 :   +COM NEW COMMENTS OF THE TABLE
  930 :   +COM <DELCO> NEW COMMENTS OF THE COLUMN
```

SQL Procedures

SQL PROCEDURES: CROSS-REFERENCES

When describing the PACKAGE (C), FUNCTION (E), PROCEDURE(Q) and TRIGGER or Rule (R) objects in the -GG associated to the object line, you can obtain automatic cross-references by using the dash character ('-'). The dash character is only considered as a logical sign and can not be used in names.

You may need:

. local data, which is prefixed by PLQ-,

. column data, which is prefixed by SLQ-.

A cross-reference can be performed on a 6-character Data Element. Longer names can however be used with a '_' character. Branching is then performed on the first part:

ex: PLQ-DATEL_COMPLEMENT.

Chapter 7. Examples of Generation Elements Screens

DB2

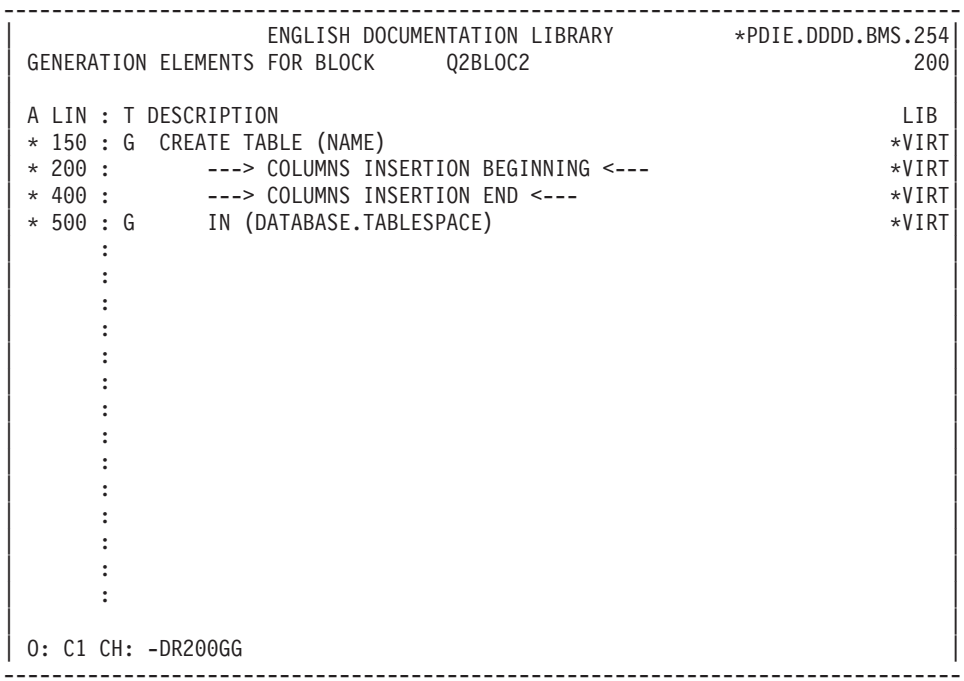
GENERATION ELEMENTS: DB2

This sub-chapter presents the Generation Elements (-GG) lines of a DB2 block ('Q2').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q2BLOC

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  CREATE DATABASE (NAME)                                       *VIRT
* 200 : G    STOGROUP ________                                          *GUID
* 300 : G  BUFFERPOOL BP0                                               *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: B q2bloc GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q2BLOC2                              080

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE TABLESPACE (NAME) IN DATABASE-NAME                  *VIRT
* 200 : G    USING STOGROUP ________                                  *GUID
* 300 : G      PRIQTY ___                                             *GUID
* 400 : G      SECQTY ___                                             *GUID
* 500 : G      ERASE  NO                                              *GUID
* 600 : G    LOCKSIZE ANY                                             *GUID
* 700 : G    BUFFERPOOL ___                                           *GUID
* 800 : G    CLOSE  YES                                               *GUID
* 900 : G    DSETPASS ________                                        *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR080GG
```


ENGLISH DOCUMENTATION LIBRARY *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK Q2BLOC2 200
A LIN : T DESCRIPTION LIB
* 150 : G CREATE TABLE (NAME) *VIRT
* 200 : ---> COLUMNS INSERTION BEGINNING <--- *VIRT
* 400 : ---> COLUMNS INSERTION END <--- *VIRT
* 500 : G IN (DATABASE.TABLESPACE) *VIRT
O: C1 CH: -DR200GG

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q2BLOC2                              130

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                    *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 450 : G      AS SELECT ALL                                          *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---           *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                 *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                              *VIRT
* 850 : G      WHERE                                                  *GUID
* 900 : G      GROUP BY                                               *GUID
* 950 : G      HAVING                                                 *GUID
* 980 : G      WITH CHECK OPTION                                      *GUID
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q2BLOC2                              210

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                                 *VIRT
* 200 : G    ON (TABLE NAME)                                            *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                    *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                          *VIRT
* 500 : G    USING STOGROUP ________                                    *GUID
* 650 : G      ERASE          NO                                        *GUID
* 700 : G    SUBPAGES         4                                         *GUID
* 750 : G    BUFFERPOOL       ___                                       *GUID
* 800 : G    CLOSE            YES                                       *GUID
* 900 : G    DSETPASS         ________                                  *GUID
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
GENERATION ELEMENTS FOR BLOCK      Q2BLOC2                              370

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  ALTER TABLE (NAME)                                           *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                    *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                          *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
*** END ***
O: C1 CH: -DR370GG
```

SQL SERVER

GENERATION ELEMENTS: SQL SERVER

This subchapter presents the Generation Elements (-GG) lines of an SQL SERVER block ('QR').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q3BLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 420 : G    ON ______                                                 *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q3BLOC                              130

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                    *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 450 : G      AS SELECT ALL                                          *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---           *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                 *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                              *VIRT
* 850 : G      WHERE                                                  *GUID
* 900 : G      WITH CHECK OPTION                                      *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      Q3BLOC                              210

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                               *VIRT
* 200 : G    ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 420 : G   WITH  FILLFACTOR = _                                      *GUID
* 440 : G    IGNORE_DUP_KEY                                           *GUID
* 450 : G    SORTED_DATA                                              *GUID
* 460 : G    IGNORE_DUP_ROW   / ALLOW_DUP_ROW                         *GUID
* 500 : G    ON  _________                                            *GUID
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR210GG
```

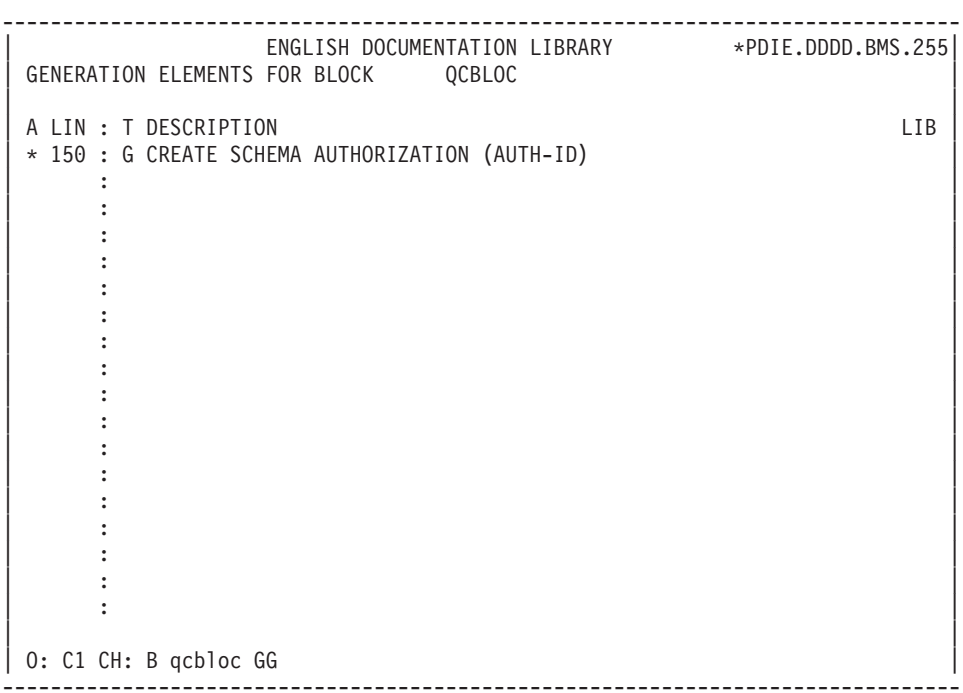
```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
GENERATION ELEMENTS FOR BLOCK      Q3BLOC                              110

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ALTER TABLE (NAME)                                          *VIRT
* 180 : G     ADD                                                      *VIRT
* 190 : G      CONSTRAINT (CONSTRAINT NAME)                            *VIRT
* 195 : G       PRIMARY KEY                                            *VIRT
* 196 : G       NONCLUSTERED                                           *GUID
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G         WITH FILLFACTOR = _                                  *GUID
* 550 : G         ON ________                                          *GUID
       :
       :
       :
       :
       :
       :
       :
       :
*** END ***
O: C1 CH: -DR110GG
```

DATACOM/DB

GENERATION ELEMENTS: DATACOM/DB

This sub-chapter presents the Generation Elements (-GG) lines of a DATACOM/DB block ('QC').

They are called by the -DRnnnGG command, nnn being a line number found in the block description (-DR).


ENGLISH DOCUMENTATION LIBRARY *PDIE.DDDD.BMS.255
GENERATION ELEMENTS FOR BLOCK QCBLOC
A LIN : T DESCRIPTION LIB
* 150 : G CREATE SCHEMA AUTHORIZATION (AUTH-ID)
O: C1 CH: B qcbloc GG

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.255
GENERATION ELEMENTS FOR BLOCK      QCBLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 500 :        PRIMARY KEY (__________)                               *VIRT
* 600 : G      IN (AREA-NAME)                                         *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.255
GENERATION ELEMENTS FOR BLOCK      QCBLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE INDEX (NAME)                                         *VIRT
* 200 :      ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.255
GENERATION ELEMENTS FOR BLOCK      QCBLOC                              130

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS SELECT ALL                                           *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
* 900 : G      GROUP BY                                                *GUID
* 950 : G      HAVING                                                  *GUID

       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
--------------------------------------------------------------------------------
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.255|
| GENERATION ELEMENTS FOR BLOCK      QCBLOC                                 700|
|                                                                              |
| A LIN : T DESCRIPTION                                                   LIB  |
| * 150 : G  ALTER TABLE (NAME)                                          *VIRT|
| * 180 : G      ADD PRIMARY KEY                                         *VIRT|
| * 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT|
| * 400 :        ---> COLUMNS INSERTION END <---                         *VIRT|
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|                                                                              |
| O: C1 CH: -DR700GG                                                           |
--------------------------------------------------------------------------------
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.255
GENERATION ELEMENTS FOR BLOCK      QCBLOC                              220

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ALTER TABLE (NAME)                                          *VIRT
* 180 : G      ADD FOREIGN KEY                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      REFERENCES (TABLE NAME)                                 *VIRT
* 700 : G      CONSTRAINT (CONSTRAINT NAME)                            *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR220GG
```

SQL/DS

GENERATION ELEMENTS: SQL/DS

This subchapter presents the Generation Elements (-GG) lines of an SQL/DS block ('QS' type).

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QSBLOC                              080

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ACQUIRE PRIVATE DBSPACE NAMED (NAME)                        *VIRT
* 200 : G    NHEADER  = ___                                            *GUID
* 300 : G    PAGES    = ___                                            *GUID
* 400 : G    PCTINDEX = ___                                            *GUID
* 500 : G    PCTFREE  = ___                                            *GUID
* 600 : G    LOCK     = ___                                            *GUID
* 700 : G    STORPOOL = ___                                            *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR080GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QSBLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 500 : G      IN (DBSPACE-NAME)                                      *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QSBLOC                              350

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS SELECT                                               *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR350GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QSBLOC                              210

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                                *VIRT
* 200 : G    ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G    PCTFREE = ___                                             *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QSBLOC                              700

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ALTER TABLE (NAME)                                          *VIRT
* 180 : G      ADD PRIMARY KEY                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR110GG
```

NONSTOP SQL

GENERATION ELEMENTS: NONSTOP SQL

This sub-chapter presents the Generation Elements (-GG) lines of a NONSTOP SQL Block ('QN').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QNBLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 500 : G      CATALOG (CATALOG-NAME)                                 *GUID
* 550 : G      ORGANIZATION KEY SEQUENCED                             *GUID
* 600 : G      PARTITION                                              *GUID
* 650 : G      SECURE                                                 *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256|
| GENERATION ELEMENTS FOR BLOCK      QNBLOC                                 130|
|                                                                              |
| A LIN : T DESCRIPTION                                                  LIB   |
| * 150 : G  CREATE VIEW (VIEW NAME)                                    *VIRT  |
| * 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT  |
| * 400 :        ---> COLUMNS INSERTION END <---                        *VIRT  |
| * 450 : G      AS SELECT ALL                                          *VIRT  |
| * 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---           *VIRT  |
| * 700 :        ---> SOURCE COLUMNS INSERTION END <---                 *VIRT  |
| * 800 : G      FROM (SOURCE TABLES NAME)                              *VIRT  |
| * 850 : G      WHERE                                                  *GUID  |
| * 900 : G      FOR PROTECTION                                         *GUID  |
| * 950 : G      CATALOG (NOM-CATALOGUE)                                *GUID  |
| * 960 : G      SECURE                                                 *GUID  |
| * 970 : G      WITH CHECK OPTION                                      *GUID  |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|                                                                              |
| O: C1 CH: -DR130GG                                                           |
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QNBLOC                              210

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                                *VIRT
* 200 : G    ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G    CATALOG (CATALOG-NAME)                                    *GUID
* 550 : G    KEYTAG (KEY-IDENTIFIER)                                   *GUID
* 600 : G    PARTITION                                                 *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
GENERATION ELEMENTS FOR BLOCK      QNBLOC                              700

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ALTER TABLE (NAME)                                          *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR700GG
```

ORACLE (V5, V6 and V7)

GENERATION ELEMENTS: ORACLE

This subchapter presents the Generation Elements (-GG) lines of an ORACLE Block (QO and QP types).

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QOBLOC                              080

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE SPACE (NAME)                                         *VIRT
* 200 : G    DATAPAGES (INITIAL __      ,INCREMENT __,                 *GUID
* 300 : G               MAXEXTENTS ____,PCTFREE __)                   *GUID
* 400 : G    INDEXPAGES INITIAL __      ,INCREMENT __,                 *GUID
* 500 : G               MAXEXTENTS ____)                              *GUID
* 600 : G    PARTITION (NAME)                                          *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR080GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QOBLOC                              100

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  CREATE TABLE (NAME)                                        *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                 *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                       *VIRT
* 500 : G      (SPACE-NAME)                                          *VIRT
* 600 : G          PCTFREE __                                        *GUID
* 650 : G          CLUSTER ________                                  *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QOBLOC                              350

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 450 : G      AS SELECT                                              *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING<---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                 *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                              *VIRT
* 850 : G      WHERE                                                  *GUID
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR350GG
```

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256|
| GENERATION ELEMENTS FOR BLOCK      QOBLOC                                 210|
|                                                                             |
| A LIN : T DESCRIPTION                                                 LIB   |
| * 150 : G  CREATE (UNIQUE) INDEX (NAME)                               *VIRT |
| * 200 : G    ON (TABLE NAME)                                          *VIRT |
| * 300 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT |
| * 400 :        ---> COLUMNS INSERTION END <---                        *VIRT |
| * 500 : G    COMPRESS                                                 *GUID |
| * 600 : G    SYSSORT                                                  *GUID |
| * 700 : G    ROWS    = ___                                            *GUID |
| * 800 : G    PCTFREE = ___                                            *GUID |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|                                                                             |
| O: C1 CH: -DR210GG                                                          |
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QOBLOC                              510

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  ALTER TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                 *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                       *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
*** END ***
O: C1 CH: -DR510GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QPBLOC                              080

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE TABLESPACE (NAME)                                    *VIRT
* 200 : G    DATAFILE ________                                         *GUID
* 300 : G      DEFAULT STORAGE ( INITIAL ____   NEXT _____            *GUID
* 310 : G                        MINEXTENTS __  MAXEXTENTS __          *GUID
* 320 : G                        PCTINCREASE __    )                   *GUID
* 400 : G      ONNLGE                                                  *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR080GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QPBLOC                              200

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 500 : G     TABLESPACE (TABLESPACE-NAME)                            *VIRT
* 600 : G      PCTFREE ___       PCTUSED  ___                         *GUID
* 650 : G      INITRANS ___      MAXTRANS ___                         *GUID
* 700 : G      STORAGE ( INITIAL ____  NEXT _____                     *GUID
* 710 : G                MINIXTENTS __ MAXEXTENTS __                  *GUID
* 720 : G                PCTINCREASE __   )                           *GUID
* 800 : G      CLUSTER _________                                      *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR200GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QPBLOC                              350

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                    *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 450 : G      AS SELECT                                              *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING<---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                 *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                              *VIRT
* 850 : G      WHERE                                                  *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR350GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      QPBLOC                              210

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                                *VIRT
* 200 : G    ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G      CLUSTER _________                                       *GUID
* 550 : G      INITRANS ___     MAXTRANS ___                           *GUID
* 600 : G     TABLESPACE (TABLESPACE NAME )                            *GUID
* 700 : G      STORAGE ( INITIAL _____    NEXT _____                  *GUID
* 710 : G                MINEXTENTS __  MAXEXTENTS __                  *GUID
* 720 : G                PCTINCREASE __    )                           *GUID
* 800 : G      PCTFREE __                                              *GUID
* 850 : G      NOSORT                                                  *GUID
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QPBLOC                              510

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  ALTER TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                 *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                       *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR510GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      V7BLOC                                 080

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  CREATE TABLESPACE (NAME)                                      *VIRT
* 200 : G    DATAFILE ________                                           *GUID
* 300 : G      DEFAULT STORAGE ( INITIAL ____   NEXT _____              *GUID
* 310 : G                        MINEXTENTS __  MAXEXTENTS __           *GUID
* 320 : G                        PCTINCREASE __    )                    *GUID
* 400 : G      ONNLGE                                                    *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR080GG
```

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257|
| GENERATION ELEMENTS FOR BLOCK      V7BLOC                                 200|
|                                                                             |
| A LIN : T DESCRIPTION                                                  LIB  |
| * 150 : G  CREATE TABLE (NAME)                                         *VIRT|
| * 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT|
| * 400 :        ---> COLUMNS INSERTION END <---                         *VIRT|
| * 500 : G     TABLESPACE (TABLESPACE-NAME)                             *VIRT|
| * 600 : G      PCTFREE ___       PCTUSED  ___                          *GUID|
| * 650 : G      INITRANS ___      MAXTRANS ___                          *GUID|
| * 700 : G      STORAGE ( INITIAL ____  NEXT _____                      *GUID|
| * 710 : G                MINIXTENTS __ MAXEXTENTS __                   *GUID|
| * 720 : G                PCTINCREASE __   )                            *GUID|
| * 800 : G      CLUSTER ________                                        *GUID|
| * 810 :        ---> NEW PARAMETERS (V7)                                *VIRT|
| * 820 : G      DISABLE CONSTRAINT ______                               *GUID|
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|                                                                             |
| O: C1 CH: -DR200GG                                                          |
```

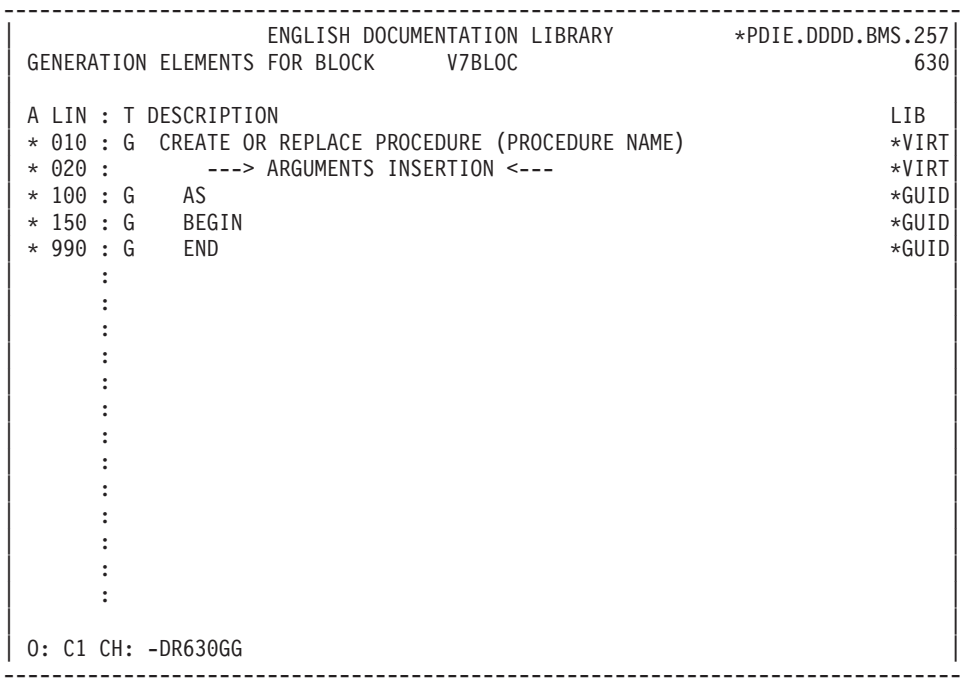
```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      V7BLOC                              350

A LIN : T DESCRIPTION                                                  LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 450 : G      AS SELECT                                              *GUID
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING<---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                 *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                              *VIRT
* 850 : G      WHERE                                                  *GUID
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR350GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.256
GENERATION ELEMENTS FOR BLOCK      V7BLOC                              210

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                               *VIRT
* 200 : G    ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G      CLUSTER _________                                       *GUID
* 550 : G      INITRANS ___     MAXTRANS ___                           *GUID
* 600 : G     TABLESPACE (TABLESPACE NAME )                           *GUID
* 700 : G      STORAGE ( INITIAL ____    NEXT _____                   *GUID
* 710 : G                MINEXTENTS __  MAXEXTENTS __                 *GUID
* 720 : G                PCTINCREASE __    )                          *GUID
* 800 : G      PCTFREE __                                              *GUID
* 850 : G      NOSORT                                                  *GUID
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      V7BLOC                              510

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  ALTER TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                 *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                       *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR510GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      V7BLOC                              560

A LIN : T DESCRIPTION                                                 LIB
* 010 : G  CREATE OR REPLACE (BODY) PACKAGE (PACKAGE NAME) AS          *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR560GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      V7BLOC                              580

A LIN : T DESCRIPTION                                                 LIB
* 010 : G  CREATE OR REPLACE FUNCTION (FUNCTION NAME)                 *VIRT
* 020 :        ---> ARGUMENTS INSERTION <---                         *VIRT
* 090 : G    RETURN _____________                                    *GUID
* 100 : G    AS                                                      *GUID
* 150 : G    BEGIN                                                   *GUID
* 990 : G    END                                                     *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR580GG
```


ENGLISH DOCUMENTATION LIBRARY *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK V7BLOC 630
A LIN : T DESCRIPTION LIB
* 010 : G CREATE OR REPLACE PROCEDURE (PROCEDURE NAME) *VIRT
* 020 : ---> ARGUMENTS INSERTION <--- *VIRT
* 100 : G AS *GUID
* 150 : G BEGIN *GUID
* 990 : G END *GUID
O: C1 CH: -DR630GG

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      V7BLOC                                 650

A LIN : T DESCRIPTION                                                    LIB
* 010 : G  CREATE TRIGGER (TRIGGER NAME)     BEFORE/AFTER                *VIRT
* 030 : G    DELETE OR INSERT OR UPDATE                                 *GUID
* 040 : G    ON (TABLE NAME)                                            *GUID
* 050 : G    REFERENCING OLD AS ______ NEW AS ______                    *GUID
* 060 : G    FOR EACH ROW                                               *GUID
* 070 : G    WHEN                                                       *GUID
* 100 : G    DECLARE                                                    *GUID
* 150 : G    BEGIN                                                      *GUID
* 990 : G    END                                                        *GUID
       :
       :
       :
       :
       :
       :
       :
       :
*** END ***
O: C1 CH: -DR650GG
```

RDMS

GENERATION ELEMENTS: RDMS

This subchapter presents the Generation Elements (-GGG) lines of an RDMS Block ('QR').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QRBLOC                              080

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE STORAGE-AREA (NAME)                                  *VIRT
* 300 : G     FOR SCHEMA      (NAME)                                   *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR080GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QRBLOC                                 100

A LIN : T DESCRIPTION                                                      LIB
* 150 : G  CREATE PERMANENT TABLE (NAME)                                  *VIRT
* 200 : G     IN              (SCHEMA NAME.STORAGE-AREA NAME)             *VIRT
* 250 :        ---> COLUMNS INSERTION BEGINNING <---                      *VIRT
* 450 :        ---> COLUMNS INSERTION END <---                            *VIRT
* 600 : G      PRIMARY KEY    (NAME)     IS (COLUMN NAME)                 *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QRBLOC                              130

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS SELECT ALL                                           *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
* 900 : G      GROUP BY                                                *GUID
* 950 : G      HAVING                                                  *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QRBLOC                              210

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE INDEX       (NAME)                                   *VIRT
* 200 : G    ON               (TABLE NAME)                             *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QRBLOC                              700

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ALTER TABLE (NAME)                                          *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR700GG
```

INTEREL RDBC

GENERATION ELEMENTS: INTEREL RDBC

This subchapter presents the Generation Elements (-GG) lines of an INTEREL RDBC Block ('QT').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QTBLOC

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  CREATE DATABASE (NAME)                                       *VIRT
* 200 : G  AS PERMANENT = ___ BYTES                                     *GUID
* 300 : G    ,SPOOL = ___ BYTES                                         *GUID
* 400 : G    ,ACCOUNT = '______'                                        *GUID
* 500 : G    ,NO FALLBACK PROTECTION                                    *GUID
* 600 : G    ,DUAL BEFORE JOURNAL                                       *GUID
* 700 : G    ,DEFAULT JOURNAL TABLE = ______                            *GUID
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QTBLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE TABLE (NAME)                                        *VIRT
* 160 : G    ,NO FALLBACK PROTECTION                                   *GUID
* 170 : G    ,DUAL BEFORE JOURNAL                                      *GUID
* 180 : G    ,WITH JOURNAL TABLE = ______                              *GUID
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G     UNIQUE PRIMARY INDEX (______)                            *GUID
* 600 : G    ,UNIQUE INDEX (______)                                    *GUID
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QTBLOC                              130

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS                                                      *VIRT
* 460 : G      LOCKING TABLE ______ FOR EXCLUSIVE MODE                 *GUID
* 480 : G      SELECT                                                  *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
      :
      :
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR130GG
```

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257|
| GENERATION ELEMENTS FOR BLOCK      QTBLOC                                 210|
|                                                                             |
| A LIN : T DESCRIPTION                                                  LIB  |
| * 150 : G  CREATE (UNIQUE) INDEX (NAME)                               *VIRT|
| * 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT|
| * 400 :        ---> COLUMNS INSERTION END <---                        *VIRT|
| * 500 : G    ON (TABLE NAME)                                          *VIRT|
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|                                                                             |
| O: C1 CH: -DR210GG                                                          |
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QTBLOC                                  700

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  ALTER TABLE (NAME)                                           *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                    *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                          *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR700GG
```

INTEREL RFM

GENERATION ELEMENTS: INTEREL RFM

This subchapter presents the Generation Elements (-GG) lines of an INTEREL RFM Block ('QU').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257|
| GENERATION ELEMENTS FOR BLOCK      QUBLOC                                 100|
|                                                                              |
| A LIN : T DESCRIPTION                                                  LIB   |
| * 150 : G  CREATE TABLE (NAME)                                        *VIRT  |
| * 160 : G  VOLUME                                                     *VIRT  |
| * 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT  |
| * 400 :        ---> COLUMNS INSERTION END <---                        *VIRT  |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|                                                                              |
| O: C1 CH: -DR100GG                                                           |
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QUBLOC                              130

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS                                                      *GUID
* 480 : G      SELECT                                                  *GUID
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257|
| GENERATION ELEMENTS FOR BLOCK      QUBLOC                                210|
|                                                                             |
| A LIN : T DESCRIPTION                                                LIB    |
| * 150 : G  CREATE (UNIQUE) INDEX (NAME)                             *VIRT   |
| * 200 : G    ON (TABLE NAME)                                        *VIRT   |
| * 300 :        ---> COLUMNS INSERTION BEGINNING <---                *VIRT   |
| * 400 :        ---> COLUMNS INSERTION END <---                      *VIRT   |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|                                                                             |
| O: C1 CH: -DR210GG                                                          |
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.257
GENERATION ELEMENTS FOR BLOCK      QUBLOC                                  700

A LIN : T DESCRIPTION                                                    LIB
* 150 : G  ALTER TABLE (NAME)                                           *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                    *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                          *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR700GG
```

DB2/2 AND DB2/6000

GENERATION ELEMENTS: DB2/2 AND DB2/6000

This subchapter presents the Generation Elements (-GG) lines of a DB2/2 or DB2/6000 Block.

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254|
| GENERATION ELEMENTS FOR BLOCK      QBBLOC                                 200|
|                                                                              |
| A LIN : T DESCRIPTION                                                  LIB   |
| * 150 : G  CREATE TABLE (NAME)                                        *VIRT  |
| * 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT  |
| * 400 :        ---> COLUMNS INSERTION END <---                        *VIRT  |
| * 500 : G      IN (DATABASE.TABLESPACE)                               *VIRT  |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|                                                                              |
| O: C1 CH: -DR200GG                                                           |
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      QBBLOC                              130

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS SELECT ALL                                           *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
* 900 : G      GROUP BY                                                *GUID
* 950 : G      HAVING                                                  *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      QBBLOC                              210

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE (UNIQUE) INDEX (NAME)                                *VIRT
* 200 : G    ON (TABLE NAME)                                           *VIRT
* 300 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 500 : G    USING STOGROUP ________                                   *GUID
* 650 : G      ERASE          NO                                       *GUID
* 700 : G    SUBPAGES         4                                        *GUID
* 750 : G    BUFFERPOOL       ___                                      *GUID
* 800 : G    CLOSE            YES                                      *GUID
* 900 : G    DSETPASS         ________                                 *GUID
      :
      :
      :
      :
      :
      :

O: C1 CH: -DR210GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
GENERATION ELEMENTS FOR BLOCK      QBBLOC                              370

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  ALTER TABLE (NAME)                                          *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: -DR370GG
```

SYBASE

GENERATION ELEMENTS: SYBASE

This subchapter presents the Generation Elements (-GG) lines of a SYBASE Block ('QR').

They are called by the -DRnnnGG command, nnn being a line number found in the Block description (-DR).

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      QYBLOC                              100

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE TABLE (NAME)                                         *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                        *VIRT
* 420 : G    ON ______                                                 *VIRT
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR100GG
```

```
                    ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254
GENERATION ELEMENTS FOR BLOCK      QYBLOC                              130

A LIN : T DESCRIPTION                                                   LIB
* 150 : G  CREATE VIEW (VIEW NAME)                                     *VIRT
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                   *VIRT
* 400 :        ---> COLUMNS INSERTION END <---                         *VIRT
* 450 : G      AS SELECT ALL                                           *VIRT
* 500 :        ---> SOURCE COLUMNS INSERTION BEGINNING <---            *VIRT
* 700 :        ---> SOURCE COLUMNS INSERTION END <---                  *VIRT
* 800 : G      FROM (SOURCE TABLES NAME)                               *VIRT
* 850 : G      WHERE                                                   *GUID
* 900 : G      WITH CHECK OPTION                                       *GUID
       :
       :
       :
       :
       :
       :
       :
       :

O: C1 CH: -DR130GG
```

```
|                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.254|
| GENERATION ELEMENTS FOR BLOCK      QYBLOC                                210|
|                                                                             |
| A LIN : T DESCRIPTION                                                  LIB  |
| * 150 : G  CREATE (UNIQUE) INDEX (NAME)                               *VIRT|
| * 200 : G    ON (TABLE NAME)                                          *VIRT|
| * 300 :        ---> COLUMNS INSERTION BEGINNING <---                  *VIRT|
| * 400 :        ---> COLUMNS INSERTION END <---                        *VIRT|
| * 420 : G   WITH  FILLFACTOR = _                                      *GUID|
| * 440 : G    IGNORE_DUP_KEY                                           *GUID|
| * 450 : G    SORTED_DATA                                              *GUID|
| * 460 : G    IGNORE_DUP_ROW   / ALLOW_DUP_ROW                         *GUID|
| * 500 : G    ON  _________                                            *GUID|
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|       :                                                                     |
|                                                                             |
| O: C1 CH: -DR210GG                                                          |
-------------------------------------------------------------------------------
```

```
                     ENGLISH DOCUMENTATION LIBRARY          *PDIE.DDDD.BMS.260
GENERATION ELEMENTS FOR BLOCK      QYBLOC                              110

A LIN : T DESCRIPTION                                                 LIB
* 150 : G  ALTER TABLE (NAME)                                        *VIRT
* 180 : G     ADD                                                    *VIRT
* 190 : G      CONSTRAINT (CONSTRAINT NAME)                          *VIRT
* 195 : G       PRIMARY KEY                                          *VIRT
* 196 : G       NONCLUSTERED                                         *GUID
* 200 :        ---> COLUMNS INSERTION BEGINNING <---                 *VIRT

* 400 :        ---> COLUMNS INSERTION END <---                       *VIRT
* 500 : G         WITH FILLFACTOR = _                                *GUID
* 550 : G         ON ________                                        *GUID
       :
       :
       :
       :
       :
       :
       :
       :
*** END ***
O: C1 CH: -DR110GG
```

Chapter 8. Block Generation

DDL Generation (B.....GN)

The DDL is generated according to the logical information specified via:

- Data Elements and Segments which define Tables, Views and Columns,
- Block Definition and Description, which define Databases, Spaces and Indexes;

and the physical information entered on the Generation Elements lines associated with the Block Definition and Description lines.

The generated DDL is displayed on-line on the SQL Commands Generation (-GN) screen, accessed via the following CHOICE:

CH: B.....GN

for the Database generation, or

CH: B.....GNnnn

for the generation of the object described on line 'nnn' of the Database Block Description (-DR) screen.

The -GNnnn screen can also be accessed by entering a 'Q' in the ACTION CODE field on the corresponding Block Description line; this Action Code cannot be used if the THE TYPE OF GENERATED TRANSACTION field (-DR screen) is blank.

On the first screen, you must indicate the TYPE and the EXTERNAL NAME (for Databases and Tablespaces); or the SEGMENT code (for Tables, Views or Indexes).

This information is automatically displayed when this screen is accessed via the CHOICE '-GNnnn' or the ACTION CODE 'Q'.

The TYPE OF GENERATED TRANSACTION value entered on the Block Description screen is also displayed.

Pressing the 'ENTER' key causes the generation of the corresponding DDL, which is then displayed on-line.

From the -GN screen, the -GN of another object may be accessed by entering:

- for Spaces:
  the TYPE and EXTERNAL NAME fields,
- for Tables, Views and Indexes:
  the TYPE and SEGMENT code fields.

For example, to obtain the -GN screen of a Table calling the Segment DDSS, you enter 'T' in the TYPE field and 'DDSS' in the SEGMENT field.

NOTE: The -GN command can only generate SQL objects PACKAGE (C), FUNCTION (E), PROCEDURE (Q), TRIGGER or RULE (R) in batch procedure.

For the generation procedure:

- local data is not taken into account.
- column data: the Data Element code is replaced by its relational name.

GENERATION OF 'ALTER' COMMANDS

ALTER TABLE commands are generated for 'I' and 'J' lines. To modify a Table, it is necessary to modify its description, i.e. the description of the corresponding Segment.

ALTER commands may only be generated through the -GN screen.

They are executed in two steps:

- Once the -GN screen corresponding to the Table is displayed, press the 'ENTER' key.
- The full description is then displayed: you must select the Table Columns for which an ALTER command is to be generated by entering a 'C' in the SELECTION (SEL.) field (see the screen examples which follow). For ORACLE, the value 'M' is used to generate 'MODIFY' commands.

  Depending on the type of line, there are three ways of specifying an ALTER TABLE command:

- ALTER TABLE on the 'I' and 'J' lines for Primary and Foreign Keys.
- ALTER TABLE on the 'A' lines for a Column update; in this case, the columns must be specified through -DR...K.
- ALTER TABLE on the 'T' lines, with the 'M' transaction code; in this case, the generation is performed in two stages.

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.202
SQL COMMANDS GENERATION     QCBLOC DATACOM/DB EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME :
TYPE       : A
             CATALOG UPDATE Y/N :

ALTER TABLE DZ10









O: C1 CH:
```

```
--------------------------------------------------------------------------------
|                    ENGLISH DOCUMENTATION LIBRARY         *PDIE.DDDD.BMS.260|
| INPUT OF DATA ELEMENTS 1    QCBLOC DATACOM/DB EXAMPLE                  700 |
|                             DZ10                                           |
|                               2    3            --------------------        |
|                               RANK ORDER       |     COLUMN        |   LIBR.|
|  COCARA                                       |   QTMLI      A    |   0067 |
|  NUCOM                        03              |   FOURNP     M    |   0067 |
|  FOURNP                       02    M         |   NUCOM           |   0067 |
|  QTMLI                        01    A         |   QTMCO      D    |   0067 |
|  QTMCO                        04    D         |   INFOR           |   0067 |
|  INFOR                        05              |                   |   0067 |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
|                                               |                   |        |
| *** END ***                                                                |
| O: C1 CH: -DR210K                                                          |
--------------------------------------------------------------------------------
```

Catalog Updating

The generated DDL may be used to update the DB2 catalog on-line.

This immediately ensures consistency between the catalog and the database description.

On the SQL Commands Generation (-GN) screen, when all input is entered, the following message is displayed:

CATALOG UPDATE Y/N:

On-line updating of the DB2 catalog may thus be requested by entering a ’Y’ (for ’YES’) in the input field following the message.

(Refer to the OPERATIONS Manual for details on the catalog updating program.)

DB2 AUTHORIZATIONS

SQL users need to be granted some authorizations for the following reasons:

1. When the user creates a Table, he is considered as its ″owner″. His code is then used as the prefix in the Table’s name.

2. The user may work on a Table he has not created, provided its owner has granted him the corresponding authorization.

There are three different kinds of Table prefixing:

A. Explicit prefixing,

B. Prefixing via the USERID,

C. Prefixing via DB2 USER.

There are two ways of generating Tables and updating the DB2 Catalog.

FIRST MODE: Batch generation via the GPRT procedure.

In cases A and B, the TSO LOGON used to update the DB2 Catalog must have authorization on the Tables corresponding to the prefix and the USERID.

In case C, the TSO LOGON is the prefix.

SECOND MODE: On-line updating.

On-line updating of the DB2 Catalog via the transaction (CICS or IMS/DC) necessitates some caution as to the organization and the granting of the System and DB2 authoriza- tions.

The user is identified differently depending on whether he works under CICS or IMS/DC and whether or not a Security system is being used. For more information please refer to the Subchapter ″TABLE OR VIEW DEFINITION″.

If the Table is prefixed (cases A and B), the user identified by DB2 must be granted the corresponding authorization by this Table's owner, i.e. via the prefix.

If the Table is not prefixed (case C), the user's identifier is used as the prefix.

Chapter 9. Access Commands

On-Line Access Commands

LISTS

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

DESCRIPTION OF BLOCK 'aaaaaa'

CHOICE	SCREEN	UPD
Baaaaaa	Definition of Database Block 'aaaaaa'	YES
BaaaaaaCR	Instances linked to Database Block 'aaaaaa' through User Relations.	YES
BaaaaaaGCbbb	Comments for Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaGGbbb	Generation Elements for Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaGObbb	Generation Options for Database Block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaATbbbbbb	Text Assigned to Database Block '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
BaaaaaaXObbbbbbCScdddd	Cross-references of Database Block 'aaaaaa' to the Call of Segments	NO

```
                  of Screen 'bbbbbb'(starting with
                  category 'c' and with Segment 'dddd').
                  Note: 'c' is equal to & for the
                  Screen-top category.

BaaaaaaXObbbbbbWccddd
                  Cross-references of Database Block     NO
                  'aaaaaa' to the Work Areas of Screen
                  'bbbbbb' (starting with Work Area 'cc',
                  line number 'ddd').

BaaaaaaXQbbbbbb   List of occurrences linked to Database NO
                  Block 'aaaaaa' through User-Defined
                  Relation (starting with Relation
                  'bbbbbb').

BaaaaaaXVvvvvvv   Cross-references of Database Block     NO
                  'aaaaaa' to Volumes (starting with
                  Volume 'vvvvvv').

BaaaaaaXPbbbbbb   Cross-references of Database Block     NO
                  'aaaaaa' to Programs (starting with
                  Program 'bbbbbb').

BaaaaaaXPbbbbbbWccddd                                    NO
                  Cross-references of Database Block
                  'aaaaaa' to Work Areas of Program
                  'bbbbbb' (starting with Work Area 'cc',
                  line number 'ddd').

LIST OF RELATIONAL/SQL OBJECTS

CHOICE            SCREEN                               UPD
------            ------                               ---

LTStddss          List of Relational/SQL Objects        NO
                  by type and code (starting with
                  with type 't', code 'ddss').

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

RELATIONAL/SQL DATABASE BLOCK DESCRIPTION

CHOICE            SCREEN                               UPD
------            ------                               ---

BaaaaaaDRbbb      Description of Relational/SQL Block  YES
                  'aaaaaa' (starting with line 'bbb').

BaaaaaaDRbbbGCccc Comments on Relational/SQL Database  YES
                  Block 'aaaaaa' description line 'bbb'
                  (starting with Comments line 'ccc').

BaaaaaaDRbbbGGccc Generation Elements of Relational/SQL YES
                  Block 'aaaaaa' description line 'bbb'
                  (starting with line 'ccc').
```

```
BaaaaaaDRbbbK      Building of Relational/SQL key 'K'    YES
                   on description line 'bbb' of Block
                   'aaaaaa'.
BaaaaaaGN          Generation of SQL commands for        YES
                   Relational/SQL block 'aaaaaa'.
BaaaaaaGNnnn       Generation of SQL commands for the    YES
                   Object defined on description line
                   'nnn' of Block 'aaaaaa'.
```

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

All notations between parentheses are optional.

CROSS-REFERENCES OF DATA ELEMENTS USED IN SQL KEYS

```
CHOICE             SCREEN                                UPD
------             ------                                ---
EaaaaaaXK          X-references of Data Element          NO
                   'aaaaaa' to the key of relational
                   /SQL database blocks.
```

LIST OF RELATIONAL INTEGRITY CONSTRAINTS

```
CHOICE             SCREEN
------             ------
SbbbbCNaaaaaa      List of integrity constraints of Segment
                   'bbbb', starting with block 'a' or 'aaa'.
```

Generation and/or Printing

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

GSQ: Generation of the DDL for the block whose code is entered in the ENTITY CODE field;

- C1 Option: Generation of Creation DDL (CREATE DATABASE and CREATE of all the objects of the block);
- C2 Option: Generation of the DDL according to the TYPE OF TRANSACTION TO GENERATE field (-DR screen).

  NOTE: ALTER commands may only be generated on-line through the -GN screen.
- C3 Option: Generation of the LABEL command for DB2, SQL/DS, and SQL400 Databases.
- C4 Option: Generation of the COMMENT command for DB2, SQL/DS, SQL400, ORACLE, and NONSTOP/SQL Databases.

  The date, time and session number will appear on the first line of the generated program.

FLS: Flow control of the block. You may specify a generation environment (PEI module), control card options, and parameters (as needed).

Chapter 10. Examples

Common Screens

This chapter shows examples of generated Blocks.

For all the Block types, you will find an example of a Table generation, of an Index generation and of a View generation.

For all the Blocks which accept it, you will also find an example of a Space generation.

This subchapter shows the screens which are common to all the Block types: the requests for a Space generation, for an Index generation and for a View generation as well as the description of the Segments used to generate these objects. As you will see in the subchapter specific to each Block type, each Block description includes three identical lines (line 100 for a Table creation, line 210 for an Index creation and line 350 for a View creation).

This subchapter shows, in the order:

- the request to generate a Table from Segment DZ05, indicated on line 100 of the Block description,
- the description of Segment DZ05,
- the request to generate an Index from Segment DZ10, indicated on line 210 of the Block description,
- the description of Segment DZ10,
- the Index composition,
- the request to generate a View from Segment DZ09, indicated on line 350 of the Block description,
- the description of Segment DZ09.

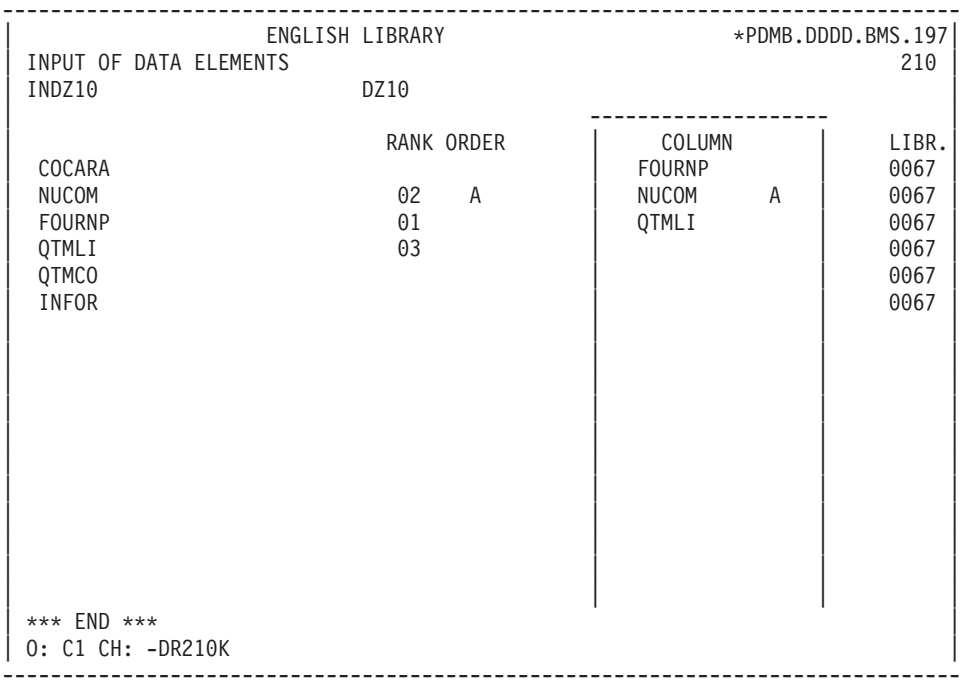
However, the results of these generation requests vary according to the Block type. So they are presented in the subchapters specific to each Block type.

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.201
SQL COMMANDS GENERATION

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T                    PREFIX :
             CREATE TABLE




*** END ***
O: C1 CH: -GN100
```

```
|                      ENGLISH LIBRARY                          *PDMB.DDDD.BMS.201|
| SEGMENT CALL OF ELEMENTS : DZ05 STOCK                                           |
|                                                                                 |
| A LIN : ELEM.  INT.FORM U OCC GR I CMD456 CONT  VALUE/SFC UPD/TRGET DOC LIBR.   |
|   100 : COCARA X        D          O                                    0067    |
|   110 : NUCOD  S9(3)    C          P         S     O                    0067    |
|   120 : FOURNI X(3)     D                                             0067    |
|   200 : NUCLIE X(8)     D                   S   O                     0067    |
|   255 : VILLE  X(20)    D        V          S   O                     0067    |
|   260 : CORESP X(256)   D        V          S      O                  0067    |
|   270 : REMISE S9(4)V99 3                                             0067    |
|   280 : MATE   X(8)     D        V                      A*MATERIEL    0067    |
|   310 : DATED  X(10)    D                                             0067    |
|   330 : HEURE  T        D                                             0067    |
|   340 : PRECIS TS       D                   S     O                   0067    |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       : NAME :                                                                  |
| *** END ***                                                                     |
| O: C2 CH: S dz05 CE                                                             |
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.197
SQL COMMANDS GENERATION

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I                    PREFIX :
             CREATE INDEX









*** END ***
O: C1 CH: -GN210
```

```
|                      ENGLISH LIBRARY                          *PDMB.DDDD.BMS.201|
| SEGMENT CALL OF ELEMENTS : DZ10 STOCK                                           |
|                                                                                 |
| A LIN : ELEM.  INT.FORM U OCC GR I CMD456 CONT  VALUE/SFC UPD/TRGET DOC LIBR.   |
|   100 : COCARA X        D          O         S  O    O                    0067|
|   110 : NUCOM  S9(5)    C          P         S  O O  O  O                 0067|
|   120 : FOURNP X(3)     D                    S  O                         0067|
|   200 : QTMLI  S9(2)    C                    S  O O     O A*LIVRABLE      0067|
|   210 : QTMCO  S9(2)    C                    S  O    O                    0067|
|   240 : INFOR  X(35)    D        V           S  O    O                    0067|
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       :                                                                         |
|       : NAME :                                                                  |
| *** END ***                                                                     |
| O: C2 CH: S dz10 CE                                                             |
```


ENGLISH LIBRARY
*PDMB.DDDD.BMS.197
INPUT OF DATA ELEMENTS
210
INDZ10
DZ10
RANK ORDER
COLUMN
LIBR.
COCARA
FOURNP
0067
NUCOM
02 A
NUCOM A
0067
FOURNP
01
QTMLI
0067
QTMLI
03
0067
QTMCO
0067
INFOR
0067
*** END ***
O: C1 CH: -DR210K

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.197
SQL COMMANDS GENERATION

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V                      PREFIX :
             CREATE VIEW
*** END ***
O: C1 CH: -GN350
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.201|
| SEGMENT CALL OF ELEMENTS : DZ09 INTERN. STOCK                                |
|                                                                              |
| A LIN : ELEM.  INT.FORM U OCC GR I CMD456 CONT VALUE/SFC UPD/TRGET DOC  LIBR.|
|   110 : FO10                 **                                      0067|
| 1 090 : CLEFO          D                                              0067|
| 1 100 : FOURNI X(3)    D       U          S    0                      0067|
| 1 110 : MATE   X(8)    D                  S 000 000000                0067|
| 1 120 : RELEA  X(3)    D       1                       LIVRABLE       0067|
| 1 130 : LANGU  X       D                                              0067|
| 1 140 : QTMAS  S9(4)   C       2                                      0067|
| 1 150 : QTMAM  S9(4)   C                                              0067|
| 1 160 : LIBFO  X(20)   D       3                                      0067|
| 1 200 : DATE   M       D                  S    0                      0067|
| 1 210 : HEURE  T       D                  S    0                      0067|
| 1 999 : FILLER XX      D                                              0067|
|   120 : COMMEN X       D                                              0067|
| * 130 : (-DBE)                                                               |
|       :                                                                      |
|       :                                                                      |
|       :                                                                      |
|       : NAME :                                                               |
|                                                                              |
| O: C2 CH: S dz09 CE                                                          |
```

DB2

DB2 EXAMPLE

The purpose of this subchapter is to show the specific screens of a DB2-type Block ('Q2BLOC' code).

You will find, in the order:

- the Block description,
- the request to generate a Space, from line 080 of the Block description,
- the result of the generation of this Space,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                          *PDMB.D801.BMS.258
RELATIONAL BLOCK DESCRIP.   Q2BLOC DB2 EXAMPLE

A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.
      :                              VIEW           TY  CDE
  080 : P ESP1-TABLE-Q2BLOC                              C              0067
  100 : T DODZ05                       DZ05              C              0067
  110 : K                              DZ05              C              0048
  130 : V VUDZ05S3                     DZ05          3   C              0058
  200 : T DODZ10                       DZ10              C              0067
  210 : I INDZ10                       DZ10              C              0048
  220 : J CEXISTF  DZ05                DZ10          C   C              0048
  300 : T                              FO10              C              0067
  350 : V VUDZ09DBE                    DZ09              C              0067
  360 : V VUDZ09S4                     DZ09          4   C              0048
  370 : A                              FO10              C              0256
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: BQ2BLOC DR
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.197
SQL COMMANDS GENERATION     Q2BLOC DB2 EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-Q2BLOC
TYPE       : P
             CREATE TABLESPACE









*** END ***
O: C1 CH: -GN080
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.197
SQL COMMANDS GENERATION     Q2BLOC DB2 EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-Q2BLOC
TYPE       : P
             CATALOG UPDATE Y/N :

CREATE TABLESPACE ESP1-TABLE-Q2BLOC           IN EXTQ2DB2
;









*** END ***
O: C1 CH:
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.197
SQL COMMANDS GENERATION     Q2BLOC DB2 EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE PDCL.DODZ05
 (COCARA             CHAR        (00001)     NOT NULL,
  NUCOD              SMALLINT                NOT NULL WITH DEFAULT,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              VARCHAR     (00020),
  CORESP             LONG VARCHAR,
  REMISE             DECIMAL     (00006, 02),
  MATERIEL           VARCHAR     (00008),
  DATED              CHAR        (00010),
  HEURE              TIME,
  PRECIS             TIMESTAMP)
   IN EXTQ2DB2.ESP1-TABLE-Q2BLOC
;

*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.197
SQL COMMANDS GENERATION     Q2BLOC DB2 EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX PDCL.INDZ10
 ON PDCL.DODZ10
(FOURNP                ,
 NUCOM              ASC ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.197|
| SQL COMMANDS GENERATION     Q2BLOC DB2 EXAMPLE                                |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW PDCL.VUDZ09DBE                                                    |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
|   LANGU             ,                                                         |
|   QTMAS             ,                                                         |
|   QTMAM             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
|  AS SELECT                                                                    |
|  PDCL.FOUR.CLEFO                                        ,                     |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
|                      ENGLISH LIBRARY                          *PDMB.DDDD.BMS.197|
| SQL COMMANDS GENERATION     Q2BLOC DB2 EXAMPLE                                   |
|                                                                                  |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                   |
| TYPE       : V                                                                   |
|              CATALOG UPDATE Y/N :                                                |
|                                                                                  |
| PDCL.FOUR.FOURNI                                          ,                      |
| PDCL.FOUR.MATE                                            ,                      |
| PDCL.FOUR.RELEA                                           ,                      |
| PDCL.FOUR.LANGU                                           ,                      |
| PDCL.FOUR.QTMAS                                           ,                      |
| PDCL.FOUR.QTMAM                                           ,                      |
| PDCL.FOUR.LIBFO                                           ,                      |
| PDCL.FOUR.DATE                                            ,                      |
| PDCL.FOUR.HEURE                                           ,                      |
| PDCL.FOUR.FILLER                                          ,                      |
| PDCL.DODZ05.COCARA                                                               |
|FROM  PDCL.FOUR                          ,                                        |
|      PDCL.DODZ05                                                                 |
|;                                                                                 |
|                                                                                  |
| *** END ***                                                                      |
| 0: C1 CH:                                                                        |
```

SQL SERVER

SQL SERVER EXAMPLE

The purpose of this subchapter is to show the specific screens of an SQL SERVER-type Block ('Q3BLOC' code).

You will find, in the order:

- the Block description,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                          *PDMB.D801.BMS.259
RELATIONAL BLOCK DESCRIP.   Q3BLOC SQL SERVER EXAMPLE

A LIN : T EXTERNAL NAME               TABLE VIEW     KEY GEN            LIBR.
      :                               CODE           TY  CDE
  100 : T DODZ05                        DZ05             C               0067
  130 : V VUDZ05S3                      DZ05         3   C               0058
  200 : T DODZ10                        DZ10             C               0067
  210 : I INDZ10                        DZ10             C               0048
  300 : T                               FO10             C               0067
  350 : V VUDZ09DBE                     DZ09             C               0067
  360 : V VUDZ09S4                      DZ09         4   C               0048
  510 : A ADDTABLE                      DZ05             C               0213
  700 : T EXTERNAL-PI00                 PI00             C               0219
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: BQ3BLOC DR
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     Q3BLOC SQL SERVER EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
 (COCARA             CHAR        (00001)     NOT NULL,
  NUCOD              SMALLINT                NULL,
  FOURNI             CHAR        (00003)     NULL,
  NUCLIE             CHAR        (00008)     NULL,
*** CORESP FIELD LENGTH >         00255 NOT TAKEN INTO ACCOUNT
  VILLE              VARCHAR     (00020)     NULL,
  REMISE             NUMERIC     (00006,02)  NULL,
  MATERIEL           VARCHAR     (00008)     NULL,
  DATED              DATETIME                NULL,
  HEURE              DATETIME                NULL,
  PRECIS             CHAR        (00026)     NULL)
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     Q3BLOC SQL SERVER EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM             ASC ,
 LIVRABLE              )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     Q3BLOC SQL SERVER EXAMPLE                          |
|                                                                                |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                  |
| TYPE       : V                                                                 |
|              CATALOG UPDATE Y/N :                                              |
|                                                                                |
| CREATE VIEW VUDZ09DBE                                                          |
|  (CLEFO             ,                                                          |
|   FOURNI            ,                                                          |
|   MATE              ,                                                          |
|   RELEA             ,                                                          |
|   LANGU             ,                                                          |
|   QTMAS             ,                                                          |
|   QTMAM             ,                                                          |
|   LIBFO             ,                                                          |
|   DATE              ,                                                          |
|   HEURE             ,                                                          |
|   FILLER            ,                                                          |
|   COMMEN            )                                                          |
|  AS SELECT                                                                     |
|  FOUR.CLEFO                          ,                                         |
| PLEASE ENTER TO CONTINUE                                                       |
| O: C1 CH:                                                                      |
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     Q3BLOC SQL SERVER EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

 FOUR.FOURNI                                              ,
 FOUR.MATE                                                ,
 FOUR.RELEA                                               ,
 FOUR.LANGU                                               ,
 FOUR.QTMAS                                               ,
 FOUR.QTMAM                                               ,
 FOUR.LIBFO                                               ,
 FOUR.DATE                                                ,
 FOUR.HEURE                                               ,
 FOUR.FILLER                                              ,
 DODZ05.COCARA
FROM  FOUR                              ,
      DODZ05
;

*** END ***
O: C1 CH:
```

DBD/2

DB2/2 EXAMPLE

The purpose of this subchapter is to show the specific screens of a DB2/2-type Block ('QBBLOC' code).

You will find, in the order:

- the Block description,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                        *PDMB.D801.BMS.259
RELATIONAL BLOCK DESCRIP.   QBBLOC DB2/2 EXAMPLE

A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.
      :                              VIEW           TY  CDE
  100 : T DODZ05                       DZ05             C              0067
  110 : K                              DZ05             C              0058
  130 : V VUDZ05S3                     DZ05         3   C              0067
  200 : T DODZ10                       DZ10             C              0048
  210 : I INDZ10                       DZ10             C              0067
  220 : J CEXISTF  DZ05                DZ10         C   C              0067
  300 : T                              FO10             C              0048
  350 : V VUDZ09DBE                    DZ09             C     +        0219
  360 : V VUDZ09S4                     DZ09         4   C
  370 : A                              FO10             C     +
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: BQBBLOC DR
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QBBLOC DB2/2 EXAMPLE                              |
|                                                                               |
| SEGMENT    : DZ05   EXTERNAL NAME : DODZ05                                    |
| TYPE       : T                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE TABLE PDCL.DODZ05                                                      |
|  (COCARA             CHAR        (00001)     NOT NULL,                        |
|   NUCOD              SMALLINT                NOT NULL WITH DEFAULT,           |
|   FOURNI             CHAR        (00003),                                     |
|   NUCLIE             CHAR        (00008),                                     |
|   VILLE              VARCHAR     (00020),                                     |
|   CORESP             VARCHAR     (00256),                                     |
|   REMISE             DECIMAL     (00006, 02),                                 |
|   MATERIEL           VARCHAR     (00008),                                     |
|   DATED              CHAR        (00010),                                     |
|   HEURE              TIME,                                                    |
|   PRECIS             TIMESTAMP)                                               |
| ;                                                                             |
|                                                                               |
|                                                                               |
| *** END ***                                                                   |
| O: C1 CH: -GN100                                                              |
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QBBLOC DB2/2 EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX PDCL.INDZ10
 ON PDCL.DODZ10
(FOURNP                 ,
 NUCOM              ASC ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QBBLOC DB2/2 EXAMPLE                              |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW PDCL.VUDZ09DBE                                                    |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
|   LANGU             ,                                                         |
|   QTMAS             ,                                                         |
|   QTMAM             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
|  AS SELECT                                                                    |
|  PDCL.FOUR.CLEFO                                        ,                     |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QBBLOC DB2/2 EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

 PDCL.FOUR.FOURNI                                        ,
 PDCL.FOUR.MATE                                          ,
 PDCL.FOUR.RELEA                                         ,
 PDCL.FOUR.LANGU                                         ,
 PDCL.FOUR.QTMAS                                         ,
 PDCL.FOUR.QTMAM                                         ,
 PDCL.FOUR.LIBFO                                         ,
 PDCL.FOUR.DATE                                          ,
 PDCL.FOUR.HEURE                                         ,
 PDCL.FOUR.FILLER                                        ,
 PDCL.DODZ05.COCARA
FROM  PDCL.FOUR                          ,
      PDCL.DODZ05
;

*** END ***
0: C1 CH:
```

DATACOM/DB

DATACOM/DB EXAMPLE

The purpose of this subchapter is to show the specific screens of a DATACOM/DB-type Block ('QCBLOC' code).

You will find, in the order:

- the Block description,
- the request to generate a Space, from line 080 of the Block description,
- the result of the generation of this Space,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
|                     ENGLISH LIBRARY                          *PDMB.D801.BMS.259|
| RELATIONAL BLOCK DESCRIP.   QCBLOC DATACOM/DB EXAMPLE                          |
|                                                                                |
| A LIN : T EXTERNAL NAME               TABLE CODE     KEY GEN            LIBR.  |
|       :                               VIEW           TY  CDE                  |
|   080 : P ESP1-TABLE-QCBLOC                              C               0067  |
|   100 : T DODZ05                        DZ05             C               0058  |
|   130 : V VUDZ05S3                      DZ05         3   C               0067  |
|   200 : T DODZ10                        DZ10             C               0067  |
|   210 : I INDZ10                        DZ10             C               0067  |
|   220 : J CEXISTF  DZ05                 DZ10             C               0048  |
|   300 : T                               FO10             C               0219  |
|   350 : V VUDZ09DBE                     DZ09             C                     |
|   360 : V VUDZ09S4                      DZ09         4   C                     |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
| *** END ***                                                                    |
| O: C1 CH: BQCBLOC DR                                                           |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QCBLOC DATACOM/DB EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QCBLOC
TYPE       : P
             CREATE SCHEMA
*** END ***
O: C1 CH: -GN080
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QCBLOC DATACOM/DB EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QCBLOC
TYPE       : P
             CATALOG UPDATE Y/N :

 CREATE SCHEMA AUTHORIZATION ESP1-TABLE-QCBLOC
 ;









*** END ***
O: C1 CH:
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QCBLOC EXEMPLE DATCOM/DB

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
 (COCARA             CHAR        (00001)     NOT NULL,
  NUCOD              SMALLINT                NOT NULL WITH DEFAULT,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              CHAR        (00020),
  CORESP             CHAR        (00256),
  REMISE             DECIMAL     (00006,02),
  MATERIEL           CHAR        (00008),
  DATED              CHAR        (00010),
  HEURE              TIME,
  PRECIS             TIMESTAMP,
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QCBLOC DATACOM/DB EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                 ,
 NUCOM                  ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QCBLOC DATACOM/DB EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

CREATE VIEW VUDZ09DBE
 (CLEFO             ,
  FOURNI            ,
  MATE              ,
  RELEA             ,
  LANGU             ,
  QTMAS             ,
  QTMAM             ,
  LIBFO             ,
  DATE              ,
  HEURE             ,
  FILLER            ,
  COMMEN            )
 AS SELECT
 FOUR.CLEFO                                              ,
PLEASE ENTER TO CONTINUE
O: C1 CH: -GN350
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QCBLOC DATACOM/DB EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

 FOUR.FOURNI                                              ,
 FOUR.MATE                                                ,
 FOUR.RELEA                                               ,
 FOUR.LANGU                                               ,
 FOUR.QTMAS                                               ,
 FOUR.QTMAM                                               ,
 FOUR.LIBFO                                               ,
 FOUR.DATE                                                ,
 FOUR.HEURE                                               ,
 FOUR.FILLER                                              ,
 DODZ05.COCARA
FROM  FOUR                              ,
      DODZ05
;

*** END ***
O: C1 CH:
```

NONSTOP SQL

NONSTOP SQL EXAMPLE

The purpose of this subchapter is to show the specific screens of a NONSTOP SQL-type Block ('QNBLOC' code).

You will find, in the order:

- the Block description,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                          *PDMB.D801.BMS.259
RELATIONAL BLOCK DESCRIP.   QNBLOC NONSTOP EXAMPLE

A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.
      :                              VIEW           TY  CDE
  100 : T DZ05                         DZ05             C              0067
  130 : V VUDZ04S3                     DZ04         3   C              0058
  200 : T DODZ10                       DZ10             C              0067
  210 : I INDZ10                       DZ10             C              0067
  300 : T                              FO10             C              0067
  350 : V VUDZ09DBE                    DZ09             C              0048
  360 : V VUDZ09S4                     DZ09         4   C              0219
  800 : T EXTERNAL-PI00                PI00             C              0219
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: BQNBLOC DR
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QNBLOC NONSTOP EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DZ05
 (COCARA             CHAR        (00001)             NOT NULL,
  NUCOD              SMALLINT               SIGNED   DEFAULT SYSTEM,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              VARCHAR     (00020),
*** REMISE INVALID USAGE                                          ***
  CORESP             VARCHAR     (00256),
  MATERIEL           VARCHAR     (00008),
  DATED              DATETIME YEAR TO HOUR,
  HEURE              TIME,
  PRECIS             TIMESTAMP)
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QNBLOC NONSTOP EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM              ASC ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QNBLOC NONSTOP EXAMPLE                            |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW VUDZ09DBE                                                         |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
|   LANGU             ,                                                         |
|   QTMAS             ,                                                         |
|   QTMAM             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
|  AS SELECT                                                                    |
|  FOUR.CLEFO                                              ,                    |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
|                      ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QNBLOC NONSTOP EXAMPLE                               |
|                                                                                  |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                   |
| TYPE       : V                                                                   |
|              CATALOG UPDATE Y/N :                                                |
|                                                                                  |
|  FOUR.FOURNI                                          ,                          |
|  FOUR.MATE                                            ,                          |
|  FOUR.RELEA                                           ,                          |
|  FOUR.LANGU                                           ,                          |
|  FOUR.QTMAS                                           ,                          |
|  FOUR.QTMAM                                           ,                          |
|  FOUR.LIBFO                                           ,                          |
|  FOUR.DATE                                            ,                          |
|  FOUR.HEURE                                           ,                          |
|  FOUR.FILLER                                          ,                          |
|  DODZ05.COCARA                                                                   |
| FROM  FOUR                          ,                                            |
|       DODZ05                                                                     |
| ;                                                                                |
|                                                                                  |
| *** END ***                                                                      |
| 0: C1 CH:                                                                        |
```

ORACLE (<V6)

ORACLE EXAMPLE

The purpose of this subchapter is to show the specific screens of an ORACLE-type Block ('QOBLOC' code).

You will find, in the order:

- the Block description,
- the request to generate a Space, from line 080 of the Block description,
- the result of the generation of this Space,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
|                     ENGLISH LIBRARY                          *PDMB.D801.BMS.259|
| RELATIONAL BLOCK DESCRIP.   QOBLOC ORACLE EXAMPLE                              |
|                                                                                |
| A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.   |
|       :                              VIEW           TY  CDE                    |
|   080 : P ESP1-TABLE-QOBLOC                              C              0067   |
|   100 : T DODZ05                       DZ05              C              0058   |
|   130 : V VUDZ05S3                     DZ05          3   C              0067   |
|   200 : T DODZ10                       DZ10              C              0067   |
|   210 : I INDZ10                       DZ10              C     +        0067   |
|   300 : T                              FO10              C              0048   |
|   350 : V VUDZ09DBE                    DZ09              C              0219   |
|   360 : V VUDZ09S4                     DZ09          4   C              0219   |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
| *** END ***                                                                    |
| O: C1 CH: BQOBLOC DR                                                           |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QOBLOC ORACLE EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QOBLOC
TYPE       : P
             CREATE SPACE
*** END ***
O: C1 CH: -GN080
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QOBLOC ORACLE EXAMPLE                             |
|                                                                               |
| SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QOBLOC                         |
| TYPE       : P                                                               |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
|  CREATE SPACE ESP1-TABLE-QOBLOC                                               |
|  ;                                                                            |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
| *** END ***                                                                   |
| O: C1 CH:                                                                     |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QOBLOC ORACLE EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
 (COCARA             CHAR        (00001)    NOT NULL,
  NUCOD              INTEGER,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              VARCHAR     (00020),
  CORESP             LONG VARCHAR,
  REMISE             NUMBER      (00006,02),
  MATERIEL           VARCHAR     (00008),
  DATED              DATE,
  HEURE              DATE,
  PRECIS             CHAR        (00026))
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QOBLOC ORACLE EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM             ASC ,
 LIVRABLE              )
;









*** END ***
O: C1 CH: -GN210
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QOBLOC ORACLE EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

CREATE VIEW VUDZ09DBE
 (CLEFO             ,
  FOURNI            ,
  MATE              ,
  RELEA             ,
  LANGU             ,
  QTMAS             ,
  QTMAM             ,
  LIBFO             ,
  DATE              ,
  HEURE             ,
  FILLER            ,
  COMMEN            )
 AS SELECT
 FOUR.CLEFO                                              ,
PLEASE ENTER TO CONTINUE
O: C1 CH: -GN350
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QOBLOC ORACLE EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

 FOUR.FOURNI                                              ,
 FOUR.MATE                                                ,
 FOUR.RELEA                                               ,
 FOUR.LANGU                                               ,
 FOUR.QTMAS                                               ,
 FOUR.QTMAM                                               ,
 FOUR.LIBFO                                               ,
 FOUR.DATE                                                ,
 FOUR.HEURE                                               ,
 FOUR.FILLER                                              ,
 DODZ05.COCARA
FROM  FOUR                              ,
      DODZ05
;

*** END ***
0: C1 CH:
```

ORACLE V7

ORACLE V7 EXAMPLE

The purpose of this subchapter is to show the specific screens of an ORACLE V7-type Block ('QPBLOC' code).

You will find, in the order:

- the Block description,
- the request to generate a Space, from line 080 of the Block description,
- the result of the generation of this Space,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.
- the request to alter a Table, from line 510 of the Block description.
- the result of the Alter Table,

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                          *PDMB.D801.BMS.259
RELATIONAL BLOCK DESCRIP.   QPBLOC ORACLE V7 EXAMPLE

A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.
      :                              VIEW           TY  CDE
  080 : P ESP1-TABLE-QPBLOC                             C              0067
  100 : T DODZ05                       DZ05             C              0058
  130 : V VUDZ05S3                     DZ05         3   C              0067
  200 : T DODZ10                       DZ10             C              0048
  210 : I INDZ10                       DZ10             C              0067
  300 : T                              FO10             C              0067
  350 : V VUDZ09DBE                    DZ09             C              0048
  360 : V VUDZ09S4                     DZ09         4   C              0213
  510 : A ADDTABLE                     DZ05             C              0219
  700 : T EXTERNAL-PI00                PI00             C    +         0219
  805 : K                              PI00             C              0219
  810 : E FONCTION                                      C              0219
      :
      :
      :
      :
*** END ***
O: C1 CH: BQPBLOC DR
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE                          |
|                                                                               |
| SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QPBLOC                         |
| TYPE       : P                                                               |
|              CREATE TABLESPACE                                                |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
| *** END ***                                                                   |
| O: C1 CH: -GN080                                                              |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QPBLOC
TYPE       : P
             CATALOG UPDATE Y/N :

CREATE TABLESPACE ESP1-TABLE-QPBLOC
;









*** END ***
O: C1 CH:
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
 (COCARA             VARCHAR2    (00001)    NOT NULL,
  NUCOD              INTEGER,
  FOURNI             VARCHAR2    (00003),
  NUCLIE             VARCHAR2    (00008),
  VILLE              VARCHAR     (00020),
  CORESP             VARCHAR     (00256),
  REMISE             NUMBER      (00006,02),
  MATERIEL           VARCHAR     (00008),
  DATED              DATE,
  HEURE              DATE,
  PRECIS             VARCHAR2    (00026))
TABLESPACE ESP1-TABLE-QPBLOC
;

*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                 ,
 NUCOM              ASC ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE                          |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW VUDZ09DBE                                                         |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
|   LANGU             ,                                                         |
|   QTMAS             ,                                                         |
|   QTMAM             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
|  AS SELECT                                                                    |
|  FOUR.CLEFO                              ,                                    |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

 FOUR.FOURNI                                                ,
 FOUR.MATE                                                  ,
 FOUR.RELEA                                                 ,
 FOUR.LANGU                                                 ,
 FOUR.QTMAS                                                 ,
 FOUR.QTMAM                                                 ,
 FOUR.LIBFO                                                 ,
 FOUR.DATE                                                  ,
 FOUR.HEURE                                                 ,
 FOUR.FILLER                                                ,
 DODZ05.COCARA
FROM  FOUR                                 ,
      DODZ05
;
                                          ,
*** END ***
O: C1 CH:
```

```
                    ENGLISH LIBRARY                        *PDMB.D801.BMS.259
SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME :
TYPE       : A
             ALTER TABLE









*** END ***
O: C1 CH: -GN510
```

```
                      ENGLISH LIBRARY                        *PDMB.D801.BMS.259
SQL COMMANDS GENERATION     QPBLOC ORACLE V7 EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME :
TYPE       : A
             CATALOG UPDATE Y/N :

ALTER TABLE DZ05









O: C1 CH:
```

RDMS

RDMS EXAMPLE

The purpose of this subchapter is to show the specific screens of an RDMS-type Block ('QRBLOC' code).

You will find, in the order:

- the Block description,
- the request to generate a Space, from line 080 of the Block description,
- the result of the generation of this Space,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
|                     ENGLISH LIBRARY                          *PDMB.D801.BMS.259|
| RELATIONAL BLOCK DESCRIP.   QRBLOC RDMS EXAMPLE                                |
|                                                                                |
| A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.   |
|       :                              VIEW           TY  CDE                    |
|   080 : P ESP1-TABLE-QRBLOC                              C              0067    |
|   100 : T DZ05                         DZ05              C              0058    |
|   200 : T DODZ10                       DZ10              C              0067    |
|   210 : I INDZ10                       DZ10              C     +        0067    |
|   300 : T                              FO10              C              0067    |
|   350 : V VUDZ09DBE                    DZ09              C              0048    |
|   360 : V VUDZ09S4                     DZ09         4    C              0219    |
|   700 : T EXTERNAL-PI00                PI00              C              0219    |
|   805 : K EXTKEY                       PI00              C     +        0219    |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
| *** END ***                                                                    |
| O: C1 CH: BQRBLOC DR                                                           |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QRBLOC RDMS EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QRBLOC
TYPE       : P
             CREATE STORAGE-AREA
*** END ***
O: C1 CH: -GN080
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QRBLOC RDMS EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QRBLOC
TYPE       : P
             CATALOG UPDATE Y/N :

 CREATE STORAGE-AREA ESP1-TABLE-QRBLOC
 FOR SCHEMA EXTQRBLC
 ;









*** END ***
O: C1 CH:
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QRBLOC RDMS EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DZ05
IN EXTQRBLC.ESP1-TABLE-QRBLOC
COLUMNS ARE COCARA             : CHARACTER   (00001)   NOT NULL  ,
            NUCOD              : DECIMAL     (00004)              ,
            FOURNI             : CHARACTER   (00003)              ,
            NUCLIE             : CHARACTER   (00008)              ,
            VILLE              : CHARACTER   (00020)              ,
            CORESP             : CHARACTER   (00256)              ,
            REMISE             : DECIMAL     (00007.02)           ,
            MATERIEL           : CHARACTER   (00008)              ,
            DATED              : CHARACTER   (00010)              ,
            HEURE              : CHARACTER   (00008)              ,
            PRECIS             : CHARACTER   (00026)
;

*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QRBLOC RDMS EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM             ASC ,
 LIVRABLE              )
;









*** END ***
O: C1 CH: -GN210
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QRBLOC RDMS EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

CREATE VIEW VUDZ09DBE
 (CLEFO             ,
  FOURNI            ,
  MATE              ,
  RELEA             ,
  LANGU             ,
  QTMAS             ,
  QTMAM             ,
  LIBFO             ,
  DATE              ,
  HEURE             ,
  FILLER            ,
  COMMEN            )
 AS SELECT
 FOUR.CLEFO                                              ,
PLEASE ENTER TO CONTINUE
O: C1 CH: -GN350
```

```
|                      ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QRBLOC RDMS EXAMPLE                                |
|                                                                                |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                  |
| TYPE       : V                                                                 |
|              CATALOG UPDATE Y/N :                                              |
|                                                                                |
|  FOUR.FOURNI                                                ,                  |
|  FOUR.MATE                                                  ,                  |
|  FOUR.RELEA                                                 ,                  |
|  FOUR.LANGU                                                 ,                  |
|  FOUR.QTMAS                                                 ,                  |
|  FOUR.QTMAM                                                 ,                  |
|  FOUR.LIBFO                                                 ,                  |
|  FOUR.DATE                                                  ,                  |
|  FOUR.HEURE                                                 ,                  |
|  FOUR.FILLER                                                ,                  |
|  DODZ05.COCARA                                                                 |
| FROM  FOUR                                ,                                    |
|       DODZ05                                                                   |
| ;                                                                              |
|                                                                                |
| *** END ***                                                                    |
| 0: C1 CH:                                                                      |
```

SQL/DS

SQL/DS EXAMPLE

The purpose of this subchapter is to show the specific screens of an SQL/DS-type Block ('QSBLOC' code).

You will find, in the order:

- the Block description,
- the request to generate a Space, from line 080 of the Block description,
- the result of the generation of this Space,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                          *PDMB.D801.BMS.259
RELATIONAL BLOCK DESCRIP.   QSBLOC SQL/DS EXAMPLE

A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.
      :                              VIEW           TY  CDE
  080 : P ESP1-TABLE-QSBLOC                              C              0067
  100 : T DODZ05                       DZ05              C              0058
  110 : K                              DZ05              C              0067
  130 : V VUDZ05S3                     DZ05          3   C              0067
  200 : T DODZ10                       DZ10              C              0067
  210 : I INDZ10                       DZ10              C     +        0048
  220 : J CEXISTF  DZ05                DZ10          C   C              0219
  300 : T                              FO10              C              0219
  350 : V VUDZ09DBE                    DZ09              C              0219
  360 : V VUDZ09S4                     DZ09          4   C
  700 : T EXTERNAL-PI00                PI00              C
  805 : K                              PI00              C     +
      :
      :
      :
*** END ***
O: C1 CH: BQSBLOC DR
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QSBLOC SQL/DS EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QSBLOC
TYPE       : P
             CREATE DBSPACE
*** END ***
O: C1 CH: -GN080
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QSBLOC SQL/DS EXAMPLE

SEGMENT    :        EXTERNAL NAME : ESP1-TABLE-QSBLOC
TYPE       : P
             CATALOG UPDATE Y/N :

 ACQUIRE PUBLIC  DBSPACE NAMED ESP1-TABLE-QSBLOC
 ;









*** END ***
O: C1 CH:
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QSBLOC SQL/DS EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE PDCL.DODZ05
 (COCARA             CHAR        (00001)     NOT NULL,
  NUCOD              SMALLINT,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              VARCHAR     (00020),
  CORESP             LONG VARCHAR,
  REMISE             DECIMAL     (00006, 02),
  MATERIEL           VARCHAR     (00008),
  DATED              CHAR        (00010),
  HEURE              TIME,
  PRECIS             TIMESTAMP)
   IN EXQSBLOC.ESP1-TABLE-QSBLOC
;

*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QSBLOC SQL/DS EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX PDCL.INDZ10
 ON PDCL.DODZ10
(FOURNP                ,
 NUCOM              ASC ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QSBLOC SQL/DS EXAMPLE                             |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW PDCL.VUDZ09DBE                                                    |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
|   LANGU             ,                                                         |
|   QTMAS             ,                                                         |
|   QTMAM             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
|  AS SELECT                                                                    |
|  PDCL.FOUR.CLEFO                                          ,                   |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
|                     ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QSBLOC SQL/DS EXAMPLE                               |
|                                                                                 |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                   |
| TYPE       : V                                                                  |
|              CATALOG UPDATE Y/N :                                               |
|                                                                                 |
|  PDCL.FOUR.FOURNI                                        ,                      |
|  PDCL.FOUR.MATE                                          ,                      |
|  PDCL.FOUR.RELEA                                         ,                      |
|  PDCL.FOUR.LANGU                                         ,                      |
|  PDCL.FOUR.QTMAS                                         ,                      |
|  PDCL.FOUR.QTMAM                                         ,                      |
|  PDCL.FOUR.LIBFO                                         ,                      |
|  PDCL.FOUR.DATE                                          ,                      |
|  PDCL.FOUR.HEURE                                         ,                      |
|  PDCL.FOUR.FILLER                                        ,                      |
|  PDCL.DODZ05.COCARA                                                             |
| FROM  PDCL.FOUR                          ,                                      |
|       PDCL.DODZ05                                                               |
| ;                                                                               |
|                                                                                 |
| *** END ***                                                                     |
| 0: C1 CH:                                                                       |
```

INTEREL RDBC

INTEREL RDBC EXAMPLE

The purpose of this subchapter is to show the specific screens of an INTEREL RDBC-type Block ('QTBLOC' code).

You will find, in the order:

- the Block description,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
|                     ENGLISH LIBRARY                          *PDMB.D801.BMS.259|
| RELATIONAL BLOCK DESCRIP.   QTBLOC INTEREL RDBC EXAMPLE                        |
|                                                                                |
| A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.   |
|       :                              VIEW           TY  CDE                   |
|   100 : T DODZ05                       DZ05              C              0067  |
|   200 : T DODZ11                       DZ11              C              0058  |
|   210 : I INDZ10                       DZ10              C     +        0067  |
|   300 : T                              FO11              C              0067  |
|   350 : V VUDZ09DBE                    DZ09              C              0067  |
|   360 : V VUDZ08S4                     DZ08         4    C              0048  |
|   800 : T EXTERNAL-PI00                PI00              C              0219  |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
| *** END ***                                                                    |
| O: C1 CH: BQTBLOC DR                                                           |
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QTBLOC INTEREL RDBC EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
/*  NUCOD  INVALID USAGE                                  */
 (COCARA             CHAR        (00001)     NOT NULL,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              VARCHAR     (00020),
/*  REMISE INVALID USAGE                                  */
  CORESP             VARCHAR     (00256),
  MATERIEL           VARCHAR     (00008),
  DATED              CHAR        (00010),
  HEURE              CHAR        (00008),
  PRECIS             CHAR        (00026))
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QTBLOC INTEREL RDBC EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM             ASC ,
 LIVRABLE              )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QTBLOC INTEREL RDBC EXAMPLE                       |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
|  CREATE VIEW VUDZ09DBE                                                        |
|   (CLEFO             ,                                                        |
|    FOURNI            ,                                                        |
|    MATE              ,                                                        |
|    RELEA             ,                                                        |
|  /*  QTMAS  INVALID USAGE                                        */           |
|  /*  QTMAM  INVALID USAGE                                        */           |
|    LANGU             ,                                                        |
|    LIBFO             ,                                                        |
|    DATE              ,                                                        |
|    HEURE             ,                                                        |
|    FILLER            ,                                                        |
|    COMMEN            )                                                        |
|  AS                                                                           |
|  SELECT                                                                       |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
|                      ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QTBLOC INTEREL RDBC EXAMPLE                        |
|                                                                                |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                  |
| TYPE       : V                                                                 |
|              CATALOG UPDATE Y/N :                                              |
|                                                                                |
|                                                                                |
|  FOUR.CLEFO                          ,                                         |
|  FOUR.FOURNI                         ,                                         |
|  FOUR.MATE                           ,                                         |
|  FOUR.RELEA                          ,                                         |
|  FOUR.LANGU                          ,                                         |
|  FOUR.LIBFO                          ,                                         |
|  FOUR.DATE                           ,                                         |
|  FOUR.HEURE                          ,                                         |
|  FOUR.FILLER                         ,                                         |
|  DODZ05.COCARA                                                                 |
|  FROM FOUR                        ,                                            |
|       DODZ05                                                                   |
| ;                                                                              |
|                                                                                |
| *** END ***                                                                    |
| O: C1 CH:                                                                      |
```

INTEREL RFM

INTEREL RFM EXAMPLE

The purpose of this subchapter is to show the specific screens of an INTEREL RFM-type Block ('QUBLOC' code).

You will find, in the order:

- the Block description,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
                    ENGLISH LIBRARY                          *PDMB.D801.BMS.259
RELATIONAL BLOCK DESCRIP.   QUBLOC INTEREL RFM EXAMPLE

A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.
      :                              VIEW           TY  CDE
  100 : T DODZ05                       DZ05             C              0067
  200 : T DODZ11                       DZ11             C              0058
  210 : I INDZ10                       DZ10             C     +        0067
  300 : T                              FO11             C              0067
  350 : V VUDZ09DBE                    DZ09             C              0067
  360 : V VUDZ08S4                     DZ08         4   C              0048
  800 : T EXTERNAL-PI00                PI00             C              0219
      :
      :
      :
      :
      :
      :
      :
      :
*** END ***
O: C1 CH: BQUBLOC DR
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QUBLOC INTEREL RFM EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
--  NUCOD  INVALID USAGE
 (COCARA             CHAR        (00001)     NOT NULL,
  FOURNI             CHAR        (00003),
  NUCLIE             CHAR        (00008),
  VILLE              CHAR        (00020),
--  REMISE INVALID USAGE
  CORESP             CHAR        (00256),
  MATERIEL           CHAR        (00008),
  DATED              CHAR        (00010),
  HEURE              CHAR        (00008),
  PRECIS             CHAR        (00026))
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QUBLOC INTEREL RFM EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM              ASC ,
 LIVRABLE               )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QUBLOC INTEREL RFM EXAMPLE                        |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW VUDZ09DBE                                                         |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
| --  QTMAS  INVALID USAGE                                                      |
| --  QTMAM  INVALID USAGE                                                      |
|   LANGU             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
| AS                                                                            |
| SELECT                                                                        |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
|                      ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QUBLOC INTEREL RFM EXAMPLE                          |
|                                                                                 |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                  |
| TYPE       : V                                                                  |
|              CATALOG UPDATE Y/N :                                               |
|                                                                                 |
|                                                                                 |
|  FOUR.CLEFO                          ,                                          |
|  FOUR.FOURNI                         ,                                          |
|  FOUR.MATE                           ,                                          |
|  FOUR.RELEA                          ,                                          |
|  FOUR.LANGU                          ,                                          |
|  FOUR.LIBFO                          ,                                          |
|  FOUR.DATE                           ,                                          |
|  FOUR.HEURE                          ,                                          |
|  FOUR.FILLER                         ,                                          |
|  DODZ05.COCARA                                                                  |
|  FROM FOUR                      ,                                               |
|       DODZ05                                                                    |
| ;                                                                               |
|                                                                                 |
| *** END ***                                                                     |
| 0: C1 CH:                                                                       |
```

SYBASE

SYBASE EXAMPLE

The purpose of this subchapter is to show the specific screens of a SYBASE-type Block ('QYBLOC' code).

You will find, in the order:

- the Block description,
- the result of a Table generation, from line 100 of the Block description,
- the result of an Index generation, from line 210 of the Block description,
- the result of a View generation, from line 350 of the Block description.

The generation requests and the Segment descriptions from which generation was performed are shown in the 'Common Screens' subchapter in this chapter.

```
|                     ENGLISH LIBRARY                          *PDMB.D801.BMS.259|
| RELATIONAL BLOCK DESCRIP.   QYBLOC SYBASE EXAMPLE                              |
|                                                                                |
| A LIN : T EXTERNAL NAME              TABLE CODE     KEY GEN            LIBR.   |
|       :                              VIEW           TY  CDE                    |
|   100 : T DODZ05                       DZ05              C              0067   |
|   130 : V VUDZ05S3                     DZ05         3    C              0058   |
|   200 : T DODZ10                       DZ10              C     +        0067   |
|   210 : I INDZ10                       DZ10              C              0067   |
|   300 : T                              FO10              C              0067   |
|   350 : V VUDZ09DBE                    DZ09              C              0048   |
|   360 : V VUDZ09S4                     DZ09         4    C              0219   |
|   510 : A ADDTABLE                     DZ05              C                     |
|   700 : T EXTERNAL-PI00                PI00              C                     |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
|       :                                                                        |
| *** END ***                                                                    |
| O: C1 CH: BQYBLOC DR                                                           |
```

```
                    ENGLISH LIBRARY                          *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QYBLOC SYBASE EXAMPLE

SEGMENT    : DZ05   EXTERNAL NAME : DODZ05
TYPE       : T
             CATALOG UPDATE Y/N :

CREATE TABLE DODZ05
 (COCARA             CHAR        (00001)     NOT NULL,
  NUCOD              SMALLINT                NULL,
  FOURNI             CHAR        (00003)     NULL,
  NUCLIE             CHAR        (00008)     NULL,
 /*  CORESP FIELD LENGTH >         00255 NOT TAKEN INTO ACCOUNT     */
  VILLE              VARCHAR     (00020)     NULL,
  REMISE             NUMERIC     (00006,02)  NULL,
  MATERIEL           VARCHAR     (00008)     NULL,
  DATED              DATETIME                NULL,
  HEURE              DATETIME                NULL,
  PRECIS             CHAR        (00026)     NULL)
;


*** END ***
O: C1 CH: -GN100
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QYBLOC SYBASE EXAMPLE

SEGMENT    : DZ10   EXTERNAL NAME : INDZ10
TYPE       : I
             CATALOG UPDATE Y/N :

CREATE INDEX INDZ10
 ON DODZ10
(FOURNP                ,
 NUCOM             ASC ,
 LIVRABLE              )
;









*** END ***
O: C1 CH: -GN210
```

```
|                     ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198|
| SQL COMMANDS GENERATION     QYBLOC SYBASE EXAMPLE                             |
|                                                                               |
| SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE                                 |
| TYPE       : V                                                                |
|              CATALOG UPDATE Y/N :                                             |
|                                                                               |
| CREATE VIEW VUDZ09DBE                                                         |
|  (CLEFO             ,                                                         |
|   FOURNI            ,                                                         |
|   MATE              ,                                                         |
|   RELEA             ,                                                         |
|   LANGU             ,                                                         |
|   QTMAS             ,                                                         |
|   QTMAM             ,                                                         |
|   LIBFO             ,                                                         |
|   DATE              ,                                                         |
|   HEURE             ,                                                         |
|   FILLER            ,                                                         |
|   COMMEN            )                                                         |
|  AS SELECT                                                                    |
|  FOUR.CLEFO                                                                   |
| PLEASE ENTER TO CONTINUE                                                      |
| O: C1 CH: -GN350                                                              |
```

```
                    ENGLISH LIBRARY                        *PDMB.DDDD.BMS.198
SQL COMMANDS GENERATION     QYBLOC SYBASE EXAMPLE

SEGMENT    : DZ09   EXTERNAL NAME : VUDZ09DBE
TYPE       : V
             CATALOG UPDATE Y/N :

 FOUR.FOURNI
 FOUR.MATE
 FOUR.RELEA
 FOUR.LANGU
 FOUR.QTMAS
 FOUR.QTMAM
 FOUR.LIBFO
 FOUR.DATE
 FOUR.HEURE
 FOUR.FILLER
 DODZ05.COCARA
FROM  FOUR                              ,
      DODZ05
;

*** END ***
O: C1 CH:
```

Part Number: DDDSQ000351A - 7421

Printed in USA