

VisualAge Pacbase



# On-Line Systems Development

*Version 3.0*

**Note**

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

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

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

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

**First Edition (December 2001)**

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.0

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

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

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

© **Copyright International Business Machines Corporation 1983,2001. All rights reserved.**

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

---

# Contents

**Notices . . . . . v**

**Trademarks. . . . . vii**

**Chapter 1. Introduction . . . . . 1**  
Purpose of the Manual . . . . . 1  
Principles of Description . . . . . 1  
The OLSD Function . . . . . 2

**Chapter 2. Basic Principles . . . . . 5**  
Characteristics of a Dialogue . . . . . 5  
Characteristics of a Screen . . . . . 5  
Generation of Maps and programs . . . . . 6

**Chapter 3. Definition and Description of  
a Dialogue or Screen . . . . . 9**  
Dialogue and Screen Definition (O.....). . . . . 9  
Screen Call of Elements (-CE) . . . . . 23  
Screen Layout (-L) . . . . . 40  
Screen Mapping (-M) . . . . . 57

**Chapter 4. Dialogue Simulation (-SIM) 75**

**Chapter 5. Description of a Transaction 79**  
Introduction . . . . . 79  
Dialogue Complement (-O) . . . . . 79  
Overriding Generated Constants (-GG) . . . . . 86  
Screen Call of Segments (-CS) . . . . . 92  
DL/1 Secondary Index Management . . . . . 100

**Chapter 6. Use of Structured Code 103**

Introduction . . . . . 103  
Specific Procedures . . . . . 103

**Chapter 7. Pacbase Web Connection  
Implementation . . . . . 107**  
Pacbase Web Connection . . . . . 107  
Structure of the Monitor. . . . . 108

**Chapter 8. Error Messages - Help  
Function . . . . . 111**  
Error Messages - Introduction . . . . . 111  
Error Messages - Coding . . . . . 112  
Overriding Automatic Error Messages . . . . . 112  
User-Defined Error Messages . . . . . 114  
Help Function - Introduction . . . . . 115  
Help Messages - Introduction . . . . . 116  
Help Messages - Coding. . . . . 117  
Coding Screen . . . . . 119  
Message Generation and/or Printing . . . . . 122  
Examples and Limitations . . . . . 122  
Description of the Error Message File . . . . . 123

**Chapter 9. Access Commands . . . . . 127**  
On-Line Access Commands . . . . . 127  
Screen Cross-References . . . . . 129  
Generation and/or Printing . . . . . 129

**Chapter 10. Appendix . . . . . 131**  
Standard Structure of the Procedure Division. . . 131  
Chart of Constants and Variables. . . . . 132



---

## Notices

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

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

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

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



---

## Trademarks

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

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

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

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

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





---

# Chapter 1. Introduction

---

## Purpose of the Manual

This manual presents the basic principles and specific characteristics that are necessary to develop on-line systems.

A dialogue is a family of screens. The On-line Screen entity is therefore the basic element of a dialogue.

Some common general functions are also presented:

- Management of application error messages,
- Management of the application 'help' function,
- Dialogue flow,
- Structured Code operators specific to the On-Line Systems Development function.

### PREREQUISITES

Refer to the 'Data Dictionary' manual for information on the following entities:

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

Refer to the 'Structured Code' manual for information on the following entities:

- Macro-Structures and their use,
- Work Areas,
- Beginning Insertions,
- Procedural Code.

Refer to the 'Character Mode User Interface' guide for information on the principles of management/exploitation of the System and the following entities and facilities:

- Texts,
- Comments,
- Keywords,
- Input Aids.

---

## 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 printed in bold italics on the screen examples which appear before the input field descriptions and allow for easy identification of a given field.

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

---

## The OLSD Function

The purpose of the On-Line Systems Development Function is to develop and generate on-line systems.

This function ensures the following:

- The description of all on-line screens using all of the advantages of the Specifications Dictionary,
- The prototyping and generation of all on-line screen layouts,
- The generation of programs, including the procedures described for each screen, as well as the screen flow of the dialogue.

The OLSD Function is supported by the following principles:

- An on-line dialogue represents interaction between screens that are logically related to each other. An on-line dialogue is therefore made up of a logical family of screens.
- An on-line dialogue is described independently of the physical characteristics of the hardware or the TP monitor in use.

**Note:** If you use Pacbase Web Connector, the characteristics of Dialogue for a communication via the Web are described in chapter 'Pacbase Web Connection'.

### GENERAL PRINCIPLES

For each dialogue, you define characteristics common to all of the screens making up the dialogue. This gives homogeneity to the screens.

Thus, all screens which make up a dialogue can default to a common presentation:

- Label types,
- Attributes,
- Number of fields per line.
- Communication fields
- Display fields (length, intensity)
- Input fields

You may override these default options for each screen of the dialogue if necessary.

The System considers each screen belonging to a dialogue as a list of fields of two types:

- Data Elements defined in the Data Dictionary,
- Specific labels of the screen.

The default options of the screen itself become the default options of the fields which make up a screen.

Each field is characterized by:

- A position on the screen (relative or absolute),
- A label, issued directly from the Data Dictionary (for the Data Elements),
- Presentation characteristics,
- Display characteristics.

You may override the default options that have been defined for each field of a dialogue or screen.

The System will automatically display the layout of each screen exactly as it will appear to the end-user.

The entire set of Segments that are necessary for the implementation of a screen are then associated with the screen. This includes all the physical characteristics (organization) and uses of each Segment (read, update, etc.).

### GENERATION

Once this data has been defined, the system ensures:

- An automatic generation of the screen layout description using the information pertaining to the physical environment and the system configuration.  
This generation of the screen description is adapted to hardware and TP monitor characteristics, and is contingent upon an option that has been defined at the library, dialogue or screen level.
- The automatic generation of a screen program which is structured into two main parts, which ensure all of the screen processing: validations and updates ('RECEPTION'), and display ('DISPLAY').
- This program will be generated in COBOL and will be automatically adapted to the hardware and TP monitor in use, contingent upon an option that has been specified at the library or dialogue level.

### THE OLSD FUNCTION SCREENS

The on-line Screen entity menu (HO) and the menu presenting the List-type screens (HOL) of this entity are displayed below.



---

## Chapter 2. Basic Principles

---

### Characteristics of a Dialogue

A Dialogue is defined by a two-character code and a name. Comment lines (-GC screen) and Keywords can be associated with a Dialogue.

The elements common to screens of a Dialogue are defined at the Dialogue level:

- Presentation options and the general characteristics of Help documentation calls are specified on the Dialogue Definition screen;
- The common files (common area, error messages file), as well as some processing options, are specified on the Dialogue Complement (-O) screen.

The common presentation options are as follows:

- The number of lines and columns on the screen;
- The type of label used for Data Elements (long, short, relational, column);
- The general presentation of these Data Elements (number of elements per line, alignment, etc.);
- The presentation attributes (intensity, color) according to the nature of the data (displayed fields, input fields, labels, error messages, etc.).

Screen branching (or screen sequence) of a dialogue is described in chapter 'Dialogue Simulation'.

---

### Characteristics of a Screen

#### DEFINITION OF A SCREEN

The On-Line Screen (O) entity is defined by a six-character code, the first two being the Dialogue code.

The general characteristics defined at the on-line dialogue level become the defaults for each screen of the dialogue. These can be overridden at the screen level.

#### DESCRIPTION OF A SCREEN

A screen is defined as an arranged list of fields of two types:

- Data Elements previously defined in the Data Dictionary,
- Complementary labels specific to a given screen.

These fields can be grouped into categories (top, bottom, repetitive); a category is specified via a special data element (not defined in the Dictionary) which is entered first before the elements of the category. The processing is executed category by category.

You can indicate the position of each field and the presentation attributes of each field on the screen.

The attributes of each field can also be overridden if you want to modify the default value that has been indicated at the dialogue or screen level.

You can also indicate the nature of a field (input, display, protected, complementary label, etc.).

'Procedure' type information can be indicated for each Data Element (validation, source field, update field, specific initialization).

The valid values for class and contents of the Data Elements are defined at the Data Dictionary level.

An elementary Data Element may be repeated horizontally and/or vertically. A group Data Element may be repeated vertically only. You can also directly enter additional labels.

Two different methods may be used in order to describe a screen:

- A description via the Screen Call of Elements (-CE) from which the Screen Layout (-L) is automatically generated.
- A description via the Screen Layout (-L) itself, or dynamically on the Screen Mapping (-M), entering the fields one-by-one, while viewing the layout as it is built.

These two methods can be used interchangeably for any one screen. For example, a screen layout can be described via the Call of Elements, and then modified on-line on the Screen Layout (-L) or Screen Mapping (-M) screens.

#### DATA ACCESS

The Screen Call of Segments (-CS), which must be accessed to process the screen (for display, validation and update), is then entered. For each Segment, physical attributes and access conditions (standard file or database type, access order, keys, etc.) are specified.

#### ADDITIONAL PROCEDURES

All of the procedures generated by the OLSD Function can be completed by procedures written in Structured Code.

See chapter 'Use of Structured Code' in the 'Structured Code' manual.

#### SCREEN BRANCHING

See chapter 'Dialogue Simulation' in this manual.

---

## **Generation of Maps and programs**

The development of an on-line system is independent from the hardware/software configuration in use. This allows the dialogue to be portable from one computer to another.

The definition of a dialogue is completed with the automatic generation of the following sub-groups:

- Screen maps,
- On-line programs,
- Screen branching.

#### SCREEN MAPS

The physical description of the screen is internal to the program for certain hardware/software configurations and is external for others. It will be generated in the screen description language of the given hardware/software configuration which can either be in native mode or in a more evolved language (BMS for IBM-CICS, MFS for IBM-IMS, FORMS for Bull...).

### ON-LINE PROGRAM

An on-line program is the succession of the following procedures:

- Reception of a message,
- Validation processing,
- Processing of received information,
- Formatting of the output,
- Display.

The On-Line Systems Development function generates all the procedures associated with a given screen. These procedures are not divided among several programs, nor are they grouped into a single program which processes more than one screen. All procedures related to a given screen are generated in a single program.

This basic system principle 'one program for one screen' has many advantages:

- Program independence vis-a-vis the structure and contents of an on-line dialogue,
- Uniform size of the on-line programs,
- Ease of maintenance,
- Ease of implementation.

The generated on-line program allows for man/machine exchanges in a pseudo-conversational mode in such a way as to optimize computer usage without penalizing the user.

Certain generated procedures are implicit to the functional description of a screen.

These are the technical procedures (reception, formulation and display of messages, initialization of fields).

Other generated procedures are implicit to the detailed analysis of an on-line program (validations and updates, data access).

A number of options taken into account at generation time (external names of standard files, terminals used, etc.) can be modified on the Dialogue or Screen 'Generation Elements' screen (-GG screen) (see chapter 'Description of a Transaction').

Finally, the procedures that are specific to a given screen, described with Structured Code, complete the preceding procedures in order to obtain the complete generation of an on-line program.

### SCREEN FLOW

The Screen Flow is the part of the screen description which 'animates' a dialogue. It ensures the branching of the different screens of a dialogue.

Screen branching is accomplished via reserved data elements (PFKEYs or characters), depending on the hardware in use. (See chapter 'Dialogue Simulation' for more details).

Writing a screen call is independent of the hardware/software combination in use.

Depending on the hardware/software combination in use, a dialogue can be generated in one of the following ways:

- Either in a unique 'monitor' program which is called by each on-line program during the dispatching phase (Example: IMS),
- Or in each of the on-line programs (Example: CICS).



---

## Chapter 3. Definition and Description of a Dialogue or Screen

---

### Dialogue and Screen Definition (O.....)

The purpose of this screen is to define either a dialogue or a screen.

#### ACCESS

The On-line Dialogue Definition Screen (O..) is accessed by entering the following in the CHOICE field (CH):

- For the On-line Dialogue Definition Screen, enter 'Odd' where 'dd' is the Dialogue code.
- For the On-line Screen Definition Screen, enter 'Oddscrn' where 'dd' is the Dialogue code and 'scrn' is the Screen code within the Dialogue.

#### GENERAL CHARACTERISTICS

On this screen, you define a Dialogue with a two- character code and an on-Line Screen with a six-character code.

The information contained on the Definition Screen provides the presentation characteristics of the screen being designed and the default values of the different screen attribute fields.

For each Library, there are defined default values which will be displayed at the Dialogue level. If these values are not suitable, they may be modified. The default values together with the values entered for the Dialogue establish the default values for each Screen of the Dialogue.

To define a Screen, only the Screen code within the Dialogue and the Dialogue or Screen name are required. Other information defaults to that specified on the Dialogue Definition Screen. Inappropriate default values can be overridden at this level. All options modified at the screen level are preceded by an asterisk '\*' when displayed on the screen.

#### PREREQUISITE DESCRIPTIONS

To define a Screen, the Dialogue to which it belongs must be defined.

#### ACTION CODES

See the 'Character Mode User Interface Guide' for the different values of the Action code.

**NOTE:** Deletion of a Dialogue or a Screen is possible only if all dependent entities (Dialogue Complements, comments, Segments etc.) have been previously deleted.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		COMPLETE CODE
1	2	DIALOGUE CODE (REQUIRED)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			To define a Dialogue, only these two characters are required.
			These are the first two characters of the complete code.
2	4		SCREEN CODE WITHIN THE DIALOGUE
			Required to define a Screen, a CS/Screen, a Business Component, a Folder, a Folder View, a Communications Monitor, etc.
			The Dialogue must have been previously defined.
3	30		DIALOGUE OR SCREEN NAME (REQUIRED IN CREAT)
			This name must be as explicit as possible.
			It allows for the automatic creation of keywords (as detailed in chapter 'Keywords', in the 'Character Mode User Interface' guide.
4	2		TYPE
			When creating an occurrence, its type is initialized with that entered at the Dialogue Definition level.
		BLANK	Standard OLSD Dialogue or Screen
		MW	Monitor for WEB communications
		C	C/S Screen (TUI client)
		MC	TUI Client Monitor
		SC	TUI Client sub-monitor
		MS	Server Monitor
		S	Business Component
		IT	Initialization/Termination Business Component
		FM	Communications Monitor
		MV	Communications Monitor / simple mode
		F	Folder
		FV	Folder view
		E	Error Message Server
		SI	Initialization server
		ST	Termination server
			NOTES ON SCREEN TYPE MODIFICATION:
			Screen types can be modified within the same category:
			. server - S, MS, SI, ST, E, FM, MV, IT
			. client - blank, C, MC, MW, SC
			NOTE : Database Administrators are allowed to modify the type by a type from the other category.
			If the screen type category is changed from server to client, the read-only fields on a server category screen are reinitialized to their default values in creation.
			CAUTION: The type of a Folder or of a Folder View cannot be changed, except when creating an occurrence and only at the Dialogue Definition level, independently of your access authorization level.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
5	2	SCREEN SIZE - NUMBER OF LINES
		PURE NUMERIC FIELD
	(1..62)	This number must be between 1 and 62 inclusive.
		Default value is 24 at the dialogue and screen level.
6	3	SCREEN SIZE - NUMBER OF COLUMNS
		PURE NUMERIC FIELD
	(1..160)	This number must be between 1 and 160 inclusive.
		Default value is 80 at the dialogue and screen level.
		The Format (number of lines and number of columns) specifies the overall dimensions of the screen.
		When screen 'X' calls screen 'Y', the size taken by screen 'Y' is the size specified on its Definition screen.
		The relative positioning of the Data Element immediately following the called screen therefore depends on the format of the called screen.
7	1	LABEL TYPE
		This option allows the choice of the label associated with each Data Element on the screen.
		At the screen level, it is used to override the global selections made at the dialogue level.
		There are four possible labels:
		.The long label (36 positions): Data Element name entered on the Data Element Definition screen,
		.The short label (18 positions), entered on the Data Element Description screen (-D),
		.The relational label (18 positions), entered on the Data Element Description (-D) screen,
		.The column heading label, entered on the Data Element Description screen (-D).
		These labels are entered on the Data Element Description (-D) screen.
	blank	Defaults to the value specified at the Dialogue level. (This option is not authorized for the Dialogue Description level).
	L	Data Element preceded by the long label (Data Element name) left-justified in 36 positions (default value at the dialogue level).
		Example:
		LABEL.....: Data Element
	M	Data Element preceded by the long label (Data Element clear name), right-justified, in 36 positions.
		Example:
		LABEL: Data Element
	N	Data Element preceded by the long label (Data Element name), left-justified with suppression of blanks on the right.
		Example :
		LABEL: Data Element

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	S	Data Element preceded by the short label, left-justified, in 18 positions.
	T	Data Element preceded by the short label, right-justified, in 18 positions.
	U	Data Element preceded by the short label, left-justified, with suppression of blanks on the right.
	R	Data Element preceded by the relational label, left-justified in 18 positions.
		NOTE: For all of the above options, the ':' character is generated (except for labels whose NATURE = 'O').
	F	Data Element only.
	1	Data Element topped by a single line column heading.
		Example:
		HEADING
		Data Element
	2	Data Element topped by a two-line column heading.
	3	Data Element topped by a three-line column heading.
		For the last three options, the longest length of the Data Element and the label is taken into consideration for relative positioning on the screen.
		NOTE: When used at the screen description level, a LABEL TYPE entered for an Element defining a screen category will cause the Data Elements within the category to default to this type.
8	2	NUMBER OF TABS PER LINE
		PURE NUMERIC FIELD For the Screen and C/S Screen entities:
		This option is used to automatically position the data elements on the screen. The tabulation points are the unseen points on the screen which divide each line in- to equal parts.
		Each Data Element for which no absolute positioning is indicated will automatically be positioned on the next available tabulation point.
		This field indicates the number of tabulation points per line. (Default value is 1 at the dialogue level).
		The maximum number authorized is 50.
9	1	INITIALIZATION CHAR. FOR VARIABLES
		Variables on the screen (Data Elements with NATURE = 'V') will be initialized with this character.
		This initial value will be positioned in the program associated with the screen.
10	2	HELP CHARACTER: SCREEN HELP
		ALL HARDWARE

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		This field may be coded with a character of your choice, left-justified and preferably only used for this purpose. For more information, see the 'Help Function' chapter in the 'On-Line Systems Development' manual specific to your environment.
		The documentation request will be automatically taken into account by the generated program.
		NOTE: If the generation type (defined on the library definition screen) is 'D' Cobol II or 85, the tests generated in function F8150 to detect the characters for documentation request are only made on the first character of the field and not on the whole field as it is the case for standard Cobol.
		HARDWARE USING PFKEYS
		Instead of entering a character in this field, users may assign a specific PFkey. (PFkeys must be coded as two characters, e.g. 05).
		NOTE: A PFkey and a character cannot be used together for documentation requests in the same dialogue.
		ON-LINE SYSTEMS DEVELOPMENT
		CICS, IMS, IBM 38/AS 400:
		If this field is coded with a specific character, numeric fields will not be generated as such at the MAP level.
		CICS:
		The following values may also be used:
		A1: PA1
		A2: PA2
		EN: 'ENTER' key
		00: 'CLEAR' key
		BULL with the C multi-screen variant:
		PFkeys may be used if the screens are of the 3270 type. In this case, they work as on an IBM hardware.
		TANDEM:
		It is not possible to use a character; only the values ranging from '01' to '32', corresponding to 'F1' to 'SF16', are taken into account.
		DEC/VAX:
		The values used for PFKEY coding are '01', '02', '03', '06' to '11' and '14' to '20'.
		MICROFOCUS and BOS/TP
		Values ranging from '01' to '24' may be used for the PFkeys.
		HP3000:
		The HELP function can only be activated by a character.
		PACBENCH CLIENT/SERVER
		CICS, IMS:
		If the field is filled in with a character, the numeric fields will not be generated as such at the MAP level.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			CICS:
			The following values are also authorized:
			'A1' for the PA1 key,
			'A2' for the PA2 key,
			'EN' for the ENTER key,
			'00' for the CLEAR key.
			TUXEDO, IMS:
			The following value is also authorized:
			'EN' for the ENTER key.
			HP3000:
			Values ranging from 1 to 8 may be used for the PFkeys.
			GCOS7, GCOS8:
			PFkeys may be used if the screens are of the 3270 type. In this case, they work as on an IBM hardware.
11	2		HELP CHARACTER: DATA ELEMENT HELP
			ALL HARDWARE
			This field may be coded with a character of your choice, left-justified and preferably only used for this purpose.
			The documentation request will be automatically taken into account in the generated program that will look for the documentation request character in the chosen Data Element.
			NOTE: If the generation type (defined on the library definition screen is 'D' Cobol II or 85, the tests generated in function F8150 to detect the characters for documentation requests are only made on the first character of the field and not on the whole field as it is the case for standard Cobol.
			HARDWARE USING PFKEYS
			Indicates the PFkey number used to call the documentation associated with a Data Element.
			The documentation request will be automatically taken into account in the generated program by positioning the cursor on the requested Data Element and pressing the PFkey defined at this level. (PFkeys must be coded as two characters, e.g. 05).
			NOTE: A PFkey and a character for documentation request cannot be used together in the same dialogue.
			If the preceding field is coded with a specific character, the data element documentation request also has to be coded with a specific character.
			Similarly, if the screen documentation is requested via a PFkey, the Data Element documentation should also be requested via a PFkey.
			ON-LINE SYSTEMS DEVELOPMENT
			CICS, IMS, IBM 38/AS 400:
			If the field is coded with a specific character, numeric fields will not be generated as such at the MAP level.
			CICS

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The following values may also be used:
		A1: PA1
		A2: PA2
		EN: 'ENTER' key
		00: 'CLEAR' key
		BULL generated with the C multi-screen variant
		PFkeys may be used on screens of the 3270 type. In this case, their use is the same as in an IBM environment.
		TANDEM
		It is not possible to use a character; only the values ranging from '01' to '32', corresponding to keys 'F1' to 'SF16', are taken into account.
		DEC/VAX
		The values used for PFKEY coding are '01', '02', '03', '06' to '11' and '14' to '20'.
		MICROFOCUS and BOS/TP
		Values ranging from '01' to '24' may be used for the PFkeys.
		HP3000
		The Help function can only be activated by a character. Since screen reception does not allow cursor positioning, it would be impossible to find which Data Element was selected.
		PACBENCH CLIENT/SERVER
		CICS, IMS
		If the field is filled in with a character, the numeric fields will not be generated as such at the MAP level.
		CICS
		The following values are also authorized:
		'A1' for the PA1 key,
		'A2' for the PA2 key,
		'EN' for the ENTER key,
		'00' for the CLEAR key.
		TUXEDO, IMS
		The following value is also authorized:
		'EN' for the ENTER key.
		HP3000
		Values ranging from 1 to 8 may be used for the PFKeys.
		GCOS7, GCOS8
		PFKeys may be used if the screens are of the 3270 type. In this case, they work as on an IBM hardware.
		ATTRIBUTES
		Select the intensity, presentation and color for five types of fields:
		Label, Display, Entry, Error message, Erroneous Field.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			These may be selected at the dialogue level and de- faulted or overridden at the screen level.
			The possible values for each attribute are the same whatever the field type.
			INTENSITY ATTRIBUTES
12	1		INTENSITY ATTRIBUTE - LABEL
		blank	Default to the Dialogue Definition.
		N	Normal intensity. (Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
13	1		INTENSITY ATTRIBUTE - DISPLAY FIELDS
		blank	Default to the Dialogue Definition.
		N	Normal intensity. (Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
14	1		INTENSITY ATTRIBUTE - INPUT FIELDS
		blank	Default to the Dialogue Definition.
		N	Normal intensity. (Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
15	1		INTENSITY ATTRIBUTE - ERROR MESSAGES
		blank	Default to the Dialogue Definition.
		N	Normal intensity. (Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
16	1		INTENSITY ATTRIBUTE ERRONEOUS FIELDS
		blank	Default to the Dialogue Definition.
		N	Normal intensity. (Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
			LAYOUT ATTRIBUTES
17	1		PRESENTATION ATTRIBUTE - LABEL
		blank	Default to the Dialogue Definition.
		N	Normal presentation. (Default value at the Dialogue level).
		B	Blinking.



NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		R	Reverse video.
		U	Underlined.
18	1		PRESENTATION ATTRIBUTE DISPLAY FIELD
		blank	Default to the Dialogue Definition.
		N	Normal presentation. (Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
19	1		PRESENTATION ATTRIBUTE INPUT FIELDS
		blank	Default to the Dialogue Definition.
		N	Normal presentation. (Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
20	1		PRESENTATION ATTRIBUTE ERROR MESSAGE
		blank	Default to the Dialogue Definition.
		N	Normal presentation. (Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
21	1		PRESENTATION ATTRIB. ERRONEOUS FIELD
		blank	Default to the Dialogue Definition.
		N	Normal presentation. (Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
			COLOR ATTRIBUTES
22	1		COLOR ATTRIBUTE - LABELS
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
23	1		COLOR ATTRIBUTE - DISPLAY FIELDS
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
24	1		COLOR ATTRIBUTE - INPUT FIELDS
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
25	1		COLOR ATTRIBUTE - ERROR MESSAGES
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
26	1		COLOR ATTRIBUTE - ERRONEOUS FIELD
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
27	1		TYPE OF COBOL TO GENERATE
		N	No generation.
			ON-LINE SYSTEMS DEVELOPMENT FUNCTION:
		X	IBM MVS COBOL II

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	O	IBM MVS COBOL/VS
	X	IBM VSE COBOL II
	1	IBM VSE COBOL/VS
	3	IBM AIX-OS/2-WINDOWS/NT COBOL
	O	IBM AS/400 COBOL
	3	MICROFOCUS COBOL
	4	BULL GCOS7 COBOL
	5	BULL GCOS8 COBOL
	6	BULL GCOS8 COBOL, TP8 Screens
	M	BULL GCOS6 DM6-TP COBOL
	7	HP-3000 COBOL
	U	UNISYS 2200 SERIES COBOL
	8	UNISYS A SERIES COBOL
	F	TANDEM COBOL
	I	DEC/VAX VMS COBOL
	K	ICL 2900 COBOL
		PACBENCH CLIENT/SERVER
	X	IBM MVS COBOL II
	O	IBM MVS COBOL/VS
		NOTE: Generation with COBOL II for occurrences belonging to the server category (S, E, FM, MV, F, FV, MS, SI, ST, IT types).
	X	IBM VSE COBOL II
	3	IBM AIX-OS/2-WINDOWS/NT COBOL
	3	MICROFOCUS COBOL
	4	BULL GCOS7 COBOL
	5	BULL GCOS8 COBOL
	7	HP-3000 COBOL
	U	UNISYS 2200 SERIES COBOL
	8	UNISYS A SERIES COBOL
	F	TANDEM COBOL
	I	DEC/VMS COBOL
	R	TUXEDO COBOL
28	1	TP MONITOR & MAP TYPE
	blank	Not valid on the Library Definition screen.
		For other screens: the default value (library value; dialogue value).
	N	No generation
		OLSD FUNCTION:
		TP Monitor COBOL
		variants
	0	CICS (IBM) (programs and BMS maps) 0, 1, X

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		IBM 36, monochrome map 2
		PC/MICROFOCUS (MS/DOS) 3
		QUESTAR 4, 5, 6
		DEC/VAX I
		UNISYS 2200 (programs and FLDP maps) U
		UNISYS Series A SDF format 8
		IBM 38 Y
		AS 400 O
		GCOS6-DTF-DFC-Vision M
		VPLUS 7
	1	IMS (IBM) (programs and MFS maps) 0, X
		IBM 36, map in color 2
		PC/MICROFOCUS (OS2) 3
		VIP TYPE 4, 5, 6
	2	CICS (IBM), BMS map in color 0, 1, X
		MICROFOCUS (UNIX) 3
	3	IMS (IBM), MFS map in color.
	4	IMS (IBM) monitor. 0, 1, X
		IBM VISUALAGE COBOL program 3
	5	CICS (IBM) monitor. 0, 1, X
	C	MULTI-TERMINAL 4, 5, 6, U
		CICS MULTI-TERMINAL 0, 1, X
		ICL K
		UNISYS Series A Logical Screen 8
	F	TDS FORMS (BULL GCOS7) 4
		DM6 TP FORMS M
		'TPSVRINIT service' type Program or
		'TPSVRDONE service' type Program R
	R	'REQUESTER' type Program F
		'CLIENT' type program R
	S	'SERVER' type program F
		or 'SERVICE' type program R
		PACBENCH CLIENT/SERVER
		TP MONITOR COBOL
		variants
	O	CICS (IBM)
		(Program and BMS map for the client) 0, X
		MICROFOCUS MS/DOS 3
		TANDEM Pathway F
		TUXEDO R

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		VPLUS 7
		UNISYS-2200 U
		UNISYS-A 8
	1	IMS
		(program and MFS map for the client) 0, X
		MICROFOCUS OS/2 3
		TANDEM TUXEDO F
		UNISYS-A Open/OLTP 8
	C	MULTI-SCREEN CICS 0, 1
		CLIENT ONLY X
		MULTI-SCREEN GCOS7 4
		MULTI-SCREEN GCOS8 5
		TUXEDO (from release 6.2 and higher) R
	2	CICS (IBM) (program and BMS color map) 0, X
		MICROFOCUS UNIX 3
	3	IMS (program and MFS color map) 0, X
		WINDOWS/NT 3
	4	IBM VISUALAGE COBOL 3
29	1	CONTROL CARDS BEFORE PROGRAMS
		Option code which selects the JCL lines to be inserted before each generated program.
	\$	No generation of program.
		Business Component / single-view (with no specification of a Folder). The value 'S' is used to disable the Business Component generation and to enable the Services Manager generation.
30	1	CONTROL CARDS AFTER PROGRAMS
		Option code which selects the JCL lines to be inserted after each generated program.
	\$	No generation of program.
31	1	CONTROL CARDS BEFORE MAP
		Screen and C/S Screen entities
		Option code that identifies the job card to be inserted before each generated Screen or C/S Screen map.
	\$	No generation of map.
		NOTE: This field is not used in a Pacbench C/S development with specification of Folder.
		Business Component / single-view (with no specification of a Folder).
		Option code which selects the JCL lines to be inserted before the generated Services Manager. The value 'S' is used to disable the generation of the Services Manager and to enable the Business Component to be generated.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
32	1		CONTROL CARDS AFTER MAP
			Screen and C/S Screen entities:
			Option code that identifies the job card to be inserted after each generated Screen or Screen c/s map.
		\$	No generation of map.
			NOTE: This field is not used in a Pacbench C/S development with the specification of Folder.
			Business Component / single-view (with no specification of Folder):
			Option code which selects the JCL lines to be inserted after the Services Manager generated.
33	8		EXTERNAL NAME OF PROGRAM
			This is the name of the program that is used in the IDENTIFICATION DIVISION of the generated program and the name of the module in the library of executable modules.
			ON-LINE SYSTEMS DEVELOPMENT:
			For a Dialogue, this field is used only if a monitor is generated.
			For a TPSVRINIT service, this field must contain value TPINIT.
			For a TOSVRDONT service, this field must contain value TPDONE.
34	8		EXT. NAME:(CS)SCREEN/SERVICES M. MAP
			ON-LINE SYSTEMS DEVELOPMENT:
			CICS: MAPSET name
			IMS: FORMAT name
			TANDEM: name of the associated SERVER for a 'R'-type screen (REQUESTER).
			BOS/TP: name of the associated SERVICE for a 'R'-type screen (CLIENT).
			name of the associated CLIENT for an 'S'-type screen (SERVICE).
			PACBENCH CLIENT/SERVER:
			CICS: name of the Mapset (client only).
			IMS: name of the format (client only).
			HP3000: external name of the FORM (client only).
			TUXEDO: name of the View (server only), except with TUXEDO 6.2 or higher (RC variant).
			In a Business Component / single-view development (with no specification of a Folder), the field must contain the external name of the Services Manager.
			The single-view development is documented in the Developer's Documentation dedicated to eBusiness Applications, in the 'Concepts - Architectures - Environments' volume.
35	8		TRANSACTION CODE
			ON-LINE SYSTEMS DEVELOPMENT:
			CICS: 4-character transaction code (default).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			IMS: transaction code associated with the dialogue.
			For IMS applications without sub-monitors, specify at the dialogue level. For IMS applications with sub-monitors, specify at the submonitor level.
			ICL: transaction code associated with the dialogue.
			TANDEM: terminal type for a REQUESTER-type screen.
			UNISYS: this field is used for screen branching.
			BOS/TP: name of the view used.
			PACBENCH CLIENT/SERVER:
			CICS/IMS: 4 character transaction code (default Dialogue transaction code).
			NOTE: If the MONIT option has been specified on the Dialogue Complements Screen, the transaction code must be indicated on the monitor or the sub-monitors depending on the architecture.
36	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: Characters bearing an accent and special characters can be declared as equivalent to an internal value in order to optimize the search of instances by keywords.
			You do that in the Administrator workbench, Users browser, Special Characters tab of the Parameters Specific Authorizations.
			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'.

---

## Screen Call of Elements (-CE)

All screen formats are made up of a group of fields having a position and a length. Some fields may contain data.

These fields are given intensity, presentation and color attributes.

There are two major types of fields in a screen:

- Fixed contents fields,
- Fields with variable contents which are either unprotected (entry fields) or protected (display fields).

The fields with variable contents correspond to Data Elements that are described and processed in the program associated with the screen layout.

The fixed contents fields are generally labels, which are directly associated with variable fields and usually specify the contents of that field.

The purpose of the Description by List is to completely describe a screen by listing the Data Elements it contains.

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

'O.....CE'.

Since 80 columns are not sufficient to allow for a complete description, there are three different views of this screen. The OPERATION field is used to distinguish the views:

- C1 - List of Data Elements with their processing options, without their attributes,
- C2 - List of Data Elements accompanied by their initial labels or presentation values, including their attributes.
- C3 - Data Element labels and Data Elements contained in the screen called (with value 'S' in the NATURE OF THE DATA ELEMENT field). Update is not allowed with this option.

#### GENERAL CHARACTERISTICS

Each variable field of a screen is defined as a Data Element (previously defined in the Data Dictionary) and is entered on the list screen for the client. For each list entry, it is required to provide:

- A line number,
- A nature (input field, protected field, etc.).

and optionally provide:

- A position on the screen,
- An option defining the associated label type,
- A set of attributes,
- The number of horizontal repetitions,
- The number of vertical repetitions,
- Associated processing options.

It is also possible to call an existing screen description in order to integrate it into another screen.

A screen description uses certain characteristics of the Data Elements' description (Name, Type, Validation, etc.). For more details, see the chapter on Data Elements in the 'Data Dictionary' manual.

The layout of the called screen can be displayed immediately (-SIM), see the following chapter.

**NOTE:** It is possible to compose the new screen directly on the layout screen, reached by using the '-L' or '-M' choice, but also on the Mapping Window of the VisualAge Pacbase WorkStation.



## THE CATEGORY CONCEPT

A category is a group of consecutive Data Elements on the list defining the Screen, but which can be positioned in any manner, corresponding to the overall logic of processing.

A screen can be made up of three categories:

- The heading of the screen (screen-top category),
- The repetitive section of the screen (repetitive category),
- The bottom of the screen (screen-bottom).

Each category is optional and can be defined only once in a given screen. Each category is delimited by a Data Element with a special nature (refer to the description in the NATURE OF THE DATA ELEMENT field).

The presentation of the screen is defined by the distribution of the Data Elements within the different categories.

The distribution also gives the processing order of the Data Elements in the generated program (screen-top Data Elements; the Data Elements from the repetitive section, as many times as they are repeated; and the screen-bottom Data Elements).

## LINE NUMBER

The LINE NUMBER is the key to each line of the screen. It is therefore possible to specify several lines associated to the same Data Element, if necessary.

However, if you enter several natures on the lines which correspond to the same Data Element, only the first nature will be taken into account.

## PREREQUISITE DESCRIPTIONS

The on-line screen must have been defined. Also, all Data Elements called in the screen must be defined in the Specifications Dictionary, except Data Elements defining categories. Short, relational or column labels, and sample values (if any) must be entered on the corresponding Data Element Description screen.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6	SCREEN CODE (REQUIRED)
		All six positions of the screen code are required. The first two characters must correspond to the code of the dialogue to which the screen belongs.
		The dialogue must have been previously defined.
2	1	ACTION CODE
	C	Creation of the line
	M	Modification of the line
	D	Deletion of the line
	A	Deletion of the line
	T	Transfer of the line
	B	Beginning of multiple deletion

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	G	Multiple transfer
	?	Request for HELP documentation
	E or -	Inhibit implicit update
	X	Implicit update without upper/lowercase processing (on certain lines only)
		On the GP-C4 screen (JCL command lines), upper/lowercase processing.
		On the GP-C1 screen, upper/lowercase processing on continuation lines only.
3	3	Line number (REQUIRED)
		PURE NUMERIC FIELD
		It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
4	6	DATA ELEMENT OR SCREEN CODE TO CALL
		ELEMENTARY DATA ELEMENT DEFINED IN THE DICTIONARY
		The default is the conversational format defined in the Dictionary. If none has been defined:
		.A Data Element displayed and protected will be in output format.
		.An unprotected Data Element will have an extended internal format, possibly including one or two additional positions for a sign and/or a decimal point.
		GROUP DATA ELEMENT DEFINING A CATEGORY
		Its definition in the dictionary is optional.
		SPECIAL DATA ELEMENTS
	ERMSG	This is used to allocate the placement of error messages. Its default length is 72 characters and it may be repeated vertically.
		It can be modified if ERMSG is defined in the Dictionary with a different format.
		Its NATURE OF THE DATA ELEMENT value must be 'P'.
	PFKEY	A programmable function key (PFkey) may be used:
		.for branching to another screen. The screen to which the dialogue branches and the PF key number are indicated in the update and source fields.
		.to initialize the screen operation code (OPER). The initialization character of the 'OPER' variable and the PFkey number are indicated in the UPDATE and SOURCE fields.
		PFKEY can be positioned anywhere on the '-CE' screen. It is processed in the generated program in:
		.F0520, in order to initialize the operation code of the screen,
		.F20, in order to ensure branching to a subsequent screen. In this case PFKEYs are processed according to the order of their input on the '-CE' screen.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The value of this function key depends on the generation variant. It is generally numerical and corresponds to the number of the keys available on the keyboard.
		CICS
		The following values may also be used:
		A1 : PA1 key
		A2 : PA2 key
		EN : ENTER key
		00 : CLEAR key
		TANDEM
		Only PFKEY values ranging from '01' to '32', corresponding to keys 'F1' to 'SF16', are taken into account.
		DEC/VAX
		The values used for PFKEY coding are '01', '02', '03', '06' to '11' and '14' to '20'.
		MICROFOCUS, BOS/TP
		Values ranging from '01' to '24' may be used for the PFkeys.
		HP3000
		Values ranging from 1 to 8 may be used for the PFkeys.
	*PASWD	WITH IMS ONLY
		The ability to set a password on a specific screen (PASSWORD macro generation at the MID level).
		This data element must be described in the Specifications Dictionary and its length cannot exceed eight characters.
		IMS : MANAGEMENT OF LOGICAL PAGING
		(For more details on the management of logical paging, refer to chapter 'Description of a Transaction', subchapter 'Dialogue Complement'.)
		In order to distinguish the different MOD LPAGEs, you must define, for the COND parameter used in the LPAGE macro, a 4-character Element which will receive the last four characters of the screen called.
		This Element must be called in the screen (its NATURE is automatically set to 'P'). Also see the description of the TYPE OF SOURCE FOR DISPLAY and SOURCE FOR DISPLAY/FIRST PART fields.
		Logical paging operator: A 5-character element must be described. Also, see the description of the TYPE OF SOURCE FOR DISPLAY and SOURCE FOR DISPLAY/FIRST PART fields.
		CALL OF ANOTHER SCREEN
		Code of the screen to be called (with value 'S' in the NATURE OF THE DATA ELEMENT field).
		If there is a screen call within the called screen, that call will be ignored.
		EXAMPLE: If screen 'Y' is called into screen 'X', any calls made by screen 'Y' are ignored.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		NAME OF A SCREEN
		If a screen code is used in this field (value 'T' in the NATURE OF THE DATA ELEMENT field), the clear name of the screen will appear. This is often used on a menu-type screen to allow you to select the screen by its title.
		FIXED LABELS
		A fixed label (NATURE OF THE DATA ELEMENT = 'L')
		may be created on the Screen Call of Elements (-CE) screen: the user must first create a line (O: C1) with NO value in this field, and enter the fixed label in the LITERALS field (O: C2).
		The characteristics of this label, apart from its value, can be modified using the 'C1' option of this screen.
		All of the characteristics can be modified on the Screen Layout (-L) or Screen Mapping (-M) screens.
		The 'slash' ('/') character serves as a delimiter at the end of the label. It is however possible to include slashes in the label. The last slash is then considered as the delimiter.
		A fixed label is limited to 30 characters, except if it is made up of only one repeated character. In this case, the limit is extended to 160 characters.
5	1	POSITIONING TYPE
	blank	RELATIVE POSITIONING
		The relative starting point is calculated from the end of the previous data element.
		The following examples show, using 'x', the first position of the relative positioning calculation for the following Data Element, with different presentation types and with repetitions of the Data Element.
		LABEL LABEL.....: DATA ELEMENTx
		DATA ELEMENTx
		LABEL LAB: ELEM(1) LAB: ELEM(1)x
		ELEM(1) ELEM(1)x LAB: ELEM(2) LAB: ELEM(2)
		ELEM(2) ELEM(2)
		ELEM(3) ELEM(3) etc.
		For screen calls, the lower right hand corner of the called screen is considered the end position.
	A	ABSOLUTE POSITIONING
		The line and Column numbers correspond to the absolute address of the Data Element on the screen.
		MODIFICATION OF POSITIONING
		Used to modify the positioning with the Screen Layout (-L).
	+	Use '+' to adjust this Data Element's previous location. The number you specify will be added to the original column or line number.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	-	Use '-' to adjust this Data Element's previous location. The number you specify will be subtracted from the original Column or line number.
		NOTE ON POSITION
		For Data Elements with column labels, the starting position is the first character of the field.
		For Data Elements with labels, the starting position is the first character of the label.
		For Data Elements without either, the starting position is the first character of the Data Element itself.
		NOTE: Elements identifying screen group categories have a position. The first Element within the category is therefore positioned relatively to the beginning of this category.
6	2	LINE NUMBER POSITIONING
		PURE NUMERIC FIELD
		IN RELATIVE POSITIONING
		This is the number of lines between two Data Elements. If the presentation option corresponds to a column heading, the number of lines in the column heading is automatically added to the indicated number at the generation level.
		IN ABSOLUTE POSITIONING
		This is the line number on the screen (checked by the System in case of overflow).
		If the line and column numbers are both equal to zero (default values), the Data Element is automatically positioned on the next available tabulation point.
7	3	COLUMN NUMBER POSITIONING
		PURE NUMERIC FIELD
		IN RELATIVE POSITIONING
		When the line number is zero, a number in this field represents the relative displacement from the previous Data Element on this line, and will be placed on this line.
		When the line number is not zero, the number in this field is the absolute column address.
		IN ABSOLUTE POSITIONING
		This is the address of the Data Element on the line.
		A zero value will correspond by default with column 2 in generation (col. 1 is reserved for the attribute). If the value is 1, the attribute is on the preceding line.
		The system detects line length overflow and field overlapping. However, it does not verify the presence of a space in front of each field.
		If the line and column numbers are both equal to zero (default values), the Data Element is automatically positioned on the next available tabulation point.
8	1	NATURE OF THE DATA ELEMENT

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		There are three different types of entities whose natures are defined.
		ELEMENTARY DATA ELEMENTS
	F	Data Element displayed and protected on the screen, but received by the program.
	P	Data Element displayed and protected on the screen, and not received by the program.
		In these two cases, if the Data Element does not have a conversational format, its output format is used.
	V	Data Element displayed, not protected on the screen (user input allowed), and received by the program.
		In this case, if the Data Element does not have a conversational format, its extended internal format is used (including sign and decimal point if relevant).
	O	Name or label of the Data Element only.
	L	Literal.
		The value of the literal is entered in the LITERALS field.
		For complete details, refer to field DATA ELEMENT OR SCREEN CODE TO CALL.
	T	Title of the screen (Clear name).
		DATA ELEMENT DEFINING A SCREEN GROUP CATEGORY
		These Data Elements separate the screen into 3 parts:
		top, repetitive or bottom.
		All Data Elements appearing after a 'group' type Data Element will belong to the category defined for the group, until the next group type Data Element or until the end of the list.
		Data Elements appearing before the first group Data Element belong to the screen top category.
	R	Beginning of the list of entities of the repetitive group.
	Z	Beginning of the list of entities of the bottom group.
		There can be a maximum of one group Data Element of each type per screen.
		Group Elements may be associated with a positioning value and a label type, which is then the default value for the elements belonging to the group.
		Repetitions may also be defined for the repetitive group.
		CALL OF A SCREEN
	S	Call of another screen
		The code of the called screen is indicated in place of the Data Element code. It will appear in the cross-references.
	W	TANDEM
		Call of an OVERLAY screen. The description of the called screen will appear in 'overlay' in the SCREEN-SECTION. You are responsible for its management.
9	1	LABEL TYPE

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		This option allows the choice of the label associated with each Data Element on the screen.
		At the screen level, it is used to override the global selections made at the dialogue level.
		There are four possible labels:
		.The long label (36 positions): Data Element name entered on the Data Element Definition screen,
		.The short label (18 positions), entered on the Data Element Description screen (-D),
		.The relational label (18 positions), entered on the Data Element Description (-D) screen,
		.The column heading label, entered on the Data Element Description screen (-D).
		These labels are entered on the Data Element Description (-D) screen.
	blank	Defaults to the value specified at the Dialogue level. (This option is not authorized for the Dialogue Description level).
	L	Data Element preceded by the long label (Data Element name) left-justified in 36 positions (default value at the dialogue level).
		Example:
		LABEL.....: Data Element
	M	Data Element preceded by the long label (Data Element clear name), right-justified, in 36 positions.
		Example:
		LABEL: Data Element
	N	Data Element preceded by the long label (Data Element name), left-justified with suppression of blanks on the right.
		Example :
		LABEL: Data Element
	S	Data Element preceded by the short label, left-justified, in 18 positions.
	T	Data Element preceded by the short label, right-justified, in 18 positions.
	U	Data Element preceded by the short label, left-justified, with suppression of blanks on the right.
	R	Data Element preceded by the relational label, left-justified in 18 positions.
		NOTE: For all of the above options, the ':' character is generated (except for labels whose NATURE = 'O').
	F	Data Element only.
	1	Data Element topped by a single line column heading.
		Example:
		HEADING
		Data Element
	2	Data Element topped by a two-line column heading.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		3	Data Element topped by a three-line column heading.
			For the last three options, the longest length of the Data Element and the label is taken into consideration for relative positioning on the screen.
			NOTE: When used at the screen description level, a LABEL TYPE entered for an Element defining a screen category will cause the Data Elements within the category to default to this type.
10	1		INTENSITY ATTRIBUTE - LABEL
		blank	Default to the Dialogue Definition.
		N	Normal intensity.
			(Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
11	1		INTENSITY ATTRIBUTE - DATA
		blank	Default to the Dialogue Definition.
		N	Normal intensity.
			(Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
12	1		PRESENTATION ATTRIBUTE - LABEL
		blank	Default to the Dialogue Definition.
		N	Normal presentation.
			(Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
13	1		PRESENTATION ATTRIBUTE - DATA
		blank	Default to the Dialogue Definition.
		N	Normal presentation.
			(Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
14	1		COLOR ATTRIBUTE - LABELS
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.



NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	T	Turquoise.
	B	Blue.
15	1	COLOR ATTRIBUTE - DATA
	blank	Default to the Dialogue selection.
	W	Normal (Default value at the Dialogue level).
	R	Red.
	P	Pink.
	Y	Yellow.
	G	Green.
	T	Turquoise.
	B	Blue.
16	1	CURSOR DEFAULT POSITION/ SKIP OPTION
		This field has a double role:
	Y	Place the cursor on this Data Element when this screen is initially invoked.
		IMS configuration:
		If this value is not entered, the cursor is positioned on the first Element of the screen; except in the case of LOGICAL PAGING (LPAGE OPTION) where the cursor is positioned at the top of the screen.
17	2	HORIZONTAL REPETITIONS
		PURE NUMERIC FIELD
		Indicates the number of occurrences of a Data Element on the same line. The label occurs only if it is a Column heading. Each occurrence of a Data Element is separated by a space. This repetition is only possible for an elementary Data Element.
		REPETITIVE CATEGORY:
		The number of horizontal repetitions of the first Data Element in the repetitive category helps determine the number of lines for each occurrence.
18	2	VERTICAL REPETITIONS
		PURE NUMERIC FIELD
		For an elementary Data Element:
		The number of vertical occurrences can be combined with the number of horizontal repetitions. The Data Element label also occurs vertically, except if it is a Column heading.
		For a repetitive category:
		All the Data Elements and labels of the repetitive part, which may contain several lines, are repeated vertically. Column headings appear only once.
19	1	PRESENCE VALIDATION OF DATA ELEMENT
		This validation is performed under the condition of the value in the VALIDATIONS CONDITIONS/SET VARIABLES field.
		Omissions to the input field will be taken as low values and converted into spaces prior to validation.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		E	Associated to PFKEY, validates the screen branching values in the generated program.
		blank	Optional Data Element.
		R	Required Data Element. It must be entered when creating a Segment, and cannot be deleted in a modification.
		S	Identical to value 'R', but the '-D' of the Data Element is not used for validations (if it appears on the first call line.)
		P	Optional Data Element. The validation is conditioned by the presence of the Data Element if the value 'P' is on the first call line of the Data Element.
			The update is conditioned by the presence of the Data Element for the Segment indicated on the call line which has the value 'P'. (UPDATE OPTION values: 'M', '+', '-').
		F	Identical to the value 'P', but the '-D' of the Data Element is not used for validations (if it is on the first call line.)
		N	MUST be entered on the Data Element's FIRST call line. This option inhibits the validations entered on the Data Element description lines. Only validations entered on the '-CE' screen are taken into account.
20	1		VALIDATION CONDITIONS/SET VARIABLES
			This field has three different uses:
			1. Data Element used as operation code for the screen.
			2. Data Element used as transaction code for the category.
			3. To specify under which conditions to execute validation and transfers of the Data Element.
			At the Screen level :
			The OLSF Function generates the automatic processing of 6 standard operations - display, update, screen branching, end of conversation, same screen, call of another screen - according to the values of an internal operation code (OPER, see the standard structure of the procedure in the Appendix of the 'On-Line Systems' manual.
		O	Operation code.
			The Data Element is identified as the external operation code. The internal operation code (OPER) will be set to the value placed in the UPDATE TARGET field when the Element takes the value entered in the SOURCE field; the UPDATE OPTION (U) field of the lines on which these values are entered must be set to 'V'.
			NOTE: 'O' may be omitted after the first of a list of operation code values is named.
			The following values may be used in the UPDATE TARGET field:
			'A' : Inquiry
			'M' : Update
			'O' : Transfer to another screen
			'P' : Redisplay

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'S' : Screen continuation (scroll)
		'E' : End
	I	Transaction Code.
		The Data Element is identified as the external transaction code. The internal transaction code (CATM) will be set to the value placed in the UPDATE TARGET field when the Element takes the value entered in the SOURCE field; the UPDATE OPTION (U) field of the lines on which these values are entered must be set to 'V'.
		The following values may be used in the UPDATE TARGET field:
		'C' : Create
		'M' : Modify
		'A' : Delete
		'X' : Create or modify as appropriate
		The Element nature must be 'V'.
	T	Validation condition.
		Validate when creating, modifying or deleting.
	X	Validation condition.
		Validate when creating or modifying.
	A	Validation and update condition.
		Validate under all conditions, but do not update the Segment.
	blank	Validation condition.
		Validate under all conditions.
21	1	UPDATE OPTION
		This field has a two-fold purpose:
		1. With Data Elements that control screen branching in a dialogue:
	S	Simulate screen branching to the screen named in the UPDATE TARGET field as specified by the Operation Code (OPER). Note: this simulation will only occur with options C1 or C3 on the Simulation (-SIM) screen.
	G	Generate (and simulate) the screen branching to the screen named in the UPDATE TARGET field as specified by the Operation Code (OPER).
		NOTE: If there is no screen named in the UPDATE TARGET field, the dialogue is ended.
		The value 'G' may be used with delayed branching (See description of the TYPE OF SOURCE FOR DISPLAY field).
		2. With Data Elements used in update:
	V	Value to validate: Comes as a complement to a Data Element's Description lines in order to indicate additional values to be validated, or to specify the internal values of the Operation Code (OPER) or transaction code (CATM) (with value 'O' or 'I' in the VALIDATION CONDITIONS/ SET VARIABLES field).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		For the Data Element PFKEY being used as the Operation Code (OPER), use the value in the UPDATE TARGET field for the value of the operation code (with value 'O' in the VALIDATION CONDITIONS/SET VARIABLES field).
	M	Update the Segment specified in the UPDATE TARGET field by substitution (MOVE). Default option.
	+	Add the contents of the Element to the Element named in the UPDATE TARGET field.
	-	Subtract the contents of the Element from the Element named in the UPDATE TARGET field, and store the result in the latter field.
		NOTE: Updates are executed only if no errors are found on the screen.
	P	Perform the standard validations on the Data Element, and then perform additional validations as indicated in the sub-function named in the UPDATE TARGET field, using the Element named in the SOURCE FOR DISPLAY field (the Element must be a standard field: the TYPE OF SOURCE FOR DISPLAY field must be left blank).
		NOTE: If there is a class or presence error, the sub-routine will not be performed. However, for Elements that fail the content validation (the valid values of the Data Element as specified on the Data Element Description Screen (-D)), the sub-routine will be performed.
22	4	UPDATE TARGET / FIRST PART
		In the generated Program associated to the Screen, this field is the Segment code to which belongs the Data Element to be updated from the Data Element in the Screen.
		In the normal case, it contains the Segment code and the Data Element (if different from -CE) to be updated.
		A Data Element declared as protected in the -CE screen of an O-entity is not moved to the Segment in the F30 function.
		Associated with UPDATE OPTION 'S' or 'G', this group Data Element contains the code of the called screen.
		Associated with UPDATE OPTION 'P', this group Data Element contains the sub-function called.
		With a 'V' UPDATE OPTION:
		. In relation to a Data Element, this field contains the value to validate, or if the value in VALIDATION CONDITIONS/SET OF VARIABLES field is 'O' or 'T', it contains the possible values of CATM or OPER.
		. In relation to the PFKEY Data Element, this field contains the possible values of the internal operation code OPER.
		The possible values of the OPER and CATM variables are described in the Appendix of this manual. It is also possible to indicate these values on the description screen of the specific Data Element.
		For further details, please refer to the 'Data Dictionary' manual, chapter 'Data Elements'.
23	6	UPDATE TARGET / LAST PART

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		(Default Option: Data Element code)
		The default option also works for a modification.
24	1	TYPE OF SOURCE FOR DISPLAY
		Specifies the contents of the source field which will be used to fill in a display Data Element.
	blank	The source field is a standard field.
	*	The source field is not a standard field. In this case, you may code any data-name in the source field, which will be taken as such in the generated program.
	D	For Data Elements used as the Operation Code (OPER), when used following a 'G' for generate in the UPDATE OPTION field, causes a delayed branching to the requested screen. The branching will occur after all validations and updating of the reception part of the program have been completed (Functions F05 through F3999-FN). The OPER value is set to 'O'.
	M	Same option as the value 'D' above, except that the OPER value is set to 'M', the OPERD value to 'O'.
	S	IMS:
		Pinpoints the logical paging operator or the field corresponding to the COND parameter in the LPAGE macro.
25	4	SOURCE FOR DISPLAY / FIRST PART
		In the generated program linked to the screen, this field corresponds to the Segment code to which the Data Element updated from the screen Data Element belongs.
	OLP	IMS: Logical paging operator.
	COND	IMS: This value specifies the field corresponding to the COND parameter used in the LPAGE macro.
		In the normal case, the code of the Segment and of the Data Element source (if different from -CE) are indicated.
		If UPDATE OPTION value = 'S' or 'G', this field contains the value of the field (or the PA key) used to call the requested screen.
		If UPDATE OPTION value = 'V' and with value 'O' or 'I' in the VALIDATION CONDITIONS/SET VARIABLES field, this field contains the value for the initialization of the internal operation code (OPER) or of the internal transaction code (CATM).
		In both cases, you do not need to enter the values with quotes, except if the value is blank.
		If UPDATE OPTION value = 'P' (PERFORM), this field contains the code of the field used to temporarily store data from the input screen field before the sub-function branching.
26	6	SOURCE FOR DISPLAY / SECOND PART
		(Default option: Data Element code)
		If the source Data Element is the same as the Data Element in the screen, this field remains blank.
27	2	LEVEL OF GENERATED SUB-FUNCTIONS
		PURE NUMERIC FIELD

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		06 to 98	The On-line Systems Development function automatically generates the following in the program associated with the screen:
			.A sub-function of Function 20 for each Data Element to be validated,
			.A sub-function of Functions 25, 35 and 60 for each Segment which is accessed in the screen.
			See the 'Structured Code' manual for the definitions of function, sub-function and level notions. Refer to subchapter 'Standard Structure of the Procedure Division' in the Appendix of the 'On-Line Systems Development' manual for the list of functions of the generated program.
			The hierarchical level is 10 by default. It can be modified for a Data Element or a Segment.
28	1		TYPE OF LITERAL
			Defines the contents of the next field, which is displayed on the Call of Elements (-CE) with OPERATION = '2' (O: C2 or I2, etc.).
		blank	The field contains a fixed label value.
		I	The field contains an initial value automatically displayed when the screen is invoked.
		P	The field contains a presentation value used for the screen simulation only.
		A	This value indicates that the following label is made up of one character repeated more than 30 times.
			INPUT EXAMPLE:
			LABEL
			T LITERALS
			A 045-
			The corresponding label is a line of 45 dashes.
			IBM 36, IBM 38, IBM AS/ 400:
		Y	This value specifies that the next field contains an INDICATOR number for attribute positioning.
29	30		DISPLAYED LITERAL
			This field contains either a literal, or an initial or presentation value. If no TYPE OF LITERAL value is entered in the preceding field, this field contains a fixed value. (see field 'DATA ELEMENT OR SCREEN CODE TO CALL')
			NOTE: The simple or double quote indicated in this area is replaced by a blank if the same quote is indicated as delimiter on the definition screen of the library. This replacement is performed in order to avoid COBOL compilation errors due to the presence of this delimiter in the 'values'.
			IBM 36, IBM 38/AS 400:
		001-099	This value is the INDICATOR number for attribute positioning on the erroneous Data Element. It is entered on the first call line of the Data Element.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		On CONTINUATION lines:
		Additional INDICATORS can be positioned. They are automatically generated in the screen's DDS and managed by Procedural Code (-P) lines. They are formatted as follows:
		IIIOTLD IIIOTLD IIIOTLD
		where:
		III= INDICATOR number (the maximum number of indicators is 99). With IBM 38 and AS/400: the value 'NUM' is used to generate numeric or signed numeric data ele.
		O = IBM 38: ON/OFF INDICATOR switch (N=OFF, SPACE=ON) IBM 36: This position must be left blank.
		T = ATTRIBUTE TYPE
		L = ATTRIBUTE FOR DATA ELEMENT LITERAL
		D = ATTRIBUTE FOR DATA ELEMENT CONTENTS
		With an indicator number:
		ATTRIBUTE TYPE CORRESPONDING ATTRIBUTE (L or D)
		I: INTENSITY
		B: BRIGHT
		D: DARK (i.e. NO DISPLAY)
		P: PRESENTATION
		B: BLINK
		R: REVERSE VIDEO
		U: UNDERLINED
		C: COLOR
		W: WHITE
		G: GREEN
		R: RED
		P: PINK
		Y: YELLOW
		T: TURQUOISE
		B: BLUE
		R: PROTECTION
		P: PROTECTED CONTENTS (in position D)
		If 'NUM' is coded:
		O = '*'
		T = 'S' or 'Y'
		If T='S', the Data Element is processed as a 'SIGNED NUMERIC' field.
		If T = 'Y', the Data Element is processed as a 'NUMERIC ONLY' field.

---

## Screen Layout (-L)

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

CH: -L

The Screen Layout (-L) screen is used:

- To directly build or modify the description of a screen, using the fields on the lower part of the screen while consulting the layout at the same time,
- To consult the screen layout to verify the position of fields and labels.

See subchapter 'Screen Call of Elements' for the definition and characteristics of the screen fields.

See chapter 'Dialogue Simulation' for the implementation of screen branching.

### NOTE

This screen is available using both OPERATION 'C1' and 'C2'. This allows you to access the 'CE' screen (O: C2) from the Layout (-L) screen.

### HOW TO MODIFY THE SCREEN DESCRIPTION

The fields at the bottom of the screen layout can be used for any type of modification to the screen description, according to the same rules found in the preceding subchapter.

Describing a screen via the Screen Layout (-L) requires the use of the PLACEMENT CHARACTER FOR REFERENCE and DELIMITER CHARACTERS OF LITERAL fields.

### PLACEMENT CHARACTER FOR REFERENCE

This field is used to:

- position (in creation) or to reposition (in modification) a field on the screen layout.
- copy a field from the layout into the input area at the bottom of the screen in order to modify its characteristics. It is also possible to directly modify the characteristics of a field without having to indicate its code, by directly referencing it on the layout.

### A - Positioning a field on the screen layout

Using the following procedure, the positioning of a field is absolute:

- ACTION CODE is 'C' or 'M' (No implicit ACTION CODE),
- LINE NUMBER (key) is entered,
- DATA ELEMENT OR SCREEN CODE TO CALL is entered,
- POSITIONING TYPE, LINE NUMBER POSITIONING and COLUMN NUMBER POSITIONING are not entered.

In the PLACEMENT field in the input area at the bottom of the screen, enter a special character which references the screen position which is to be used to 'position' the given field. (This character is in column position 'minus one' of the field).



EXAMPLE:

To position the Data Element 'NUTOT' (Format = ZZ9.99, label = NEW TOTAL) on line 2, column 8 of the layout, enter the following:

Step 1.

```
      7
-----
!
2 !      *
!
!
! 0: C1 CH:      ACTION:C LINE NU.: 100   PLC: * DLM:
! ETC.
-----
```

Here, the character (in this case, an asterisk) in position 2/7 of the screen layout matches the one in the PLACEMENT CHARACTER FOR REFERENCE.

The result will be one of the following screens depending on the Data Element's NATURE:

```
      8
-----
!
2 !      NEW TOTAL : ZZ9.9
!
!
! 0: C1 CH:      ACTION:C LINE NU.: 100   PLC: * DLM:
! ETC.
-----
```

OR:

```
      8
-----
!
2 !      NEW TOTAL :
!      ZZ9.9
!
!
! 0: C1 CH:      ACTION:C LINE NU.: 100   PLC: * DLM:
! ETC.
-----
```

OR:

```
      8
-----
!
2 !      ZZ9.9
!
!
! 0: C1 CH:      ACTION:C LINE NU.: 100   PLC: * DLM:
! ETC.
-----
```

This same procedure is used for a modification (ACTION CODE value = 'M').

B - Referencing a field on the Screen Layout

- ACTION CODE is 'E' or 'M'. (ACTION CODE 'E' for extraction exists in this specific case).
- LINE NUMBER (key) is NOT entered.
- DATA ELEMENT OR SCREEN CODE TO CALL is NOT entered.

- POSITIONING TYPE, LINE NUMBER POSITIONING and COLUMN NUMBER POSITIONING are NOT entered.

In the PLACEMENT field, you indicate a character found in the field to be referenced on the Layout (this character can either be in the label or in the data element contents).

To copy or to modify a field:

```

      8
-----
!
2 !      NEW TOTAL : ZZ9.9      !
!
!
! 0: C1 CH:      ACTION:E LINE NU.:      PLC: W DLM:      !
! ETC.      !
-----

```

VA Pac will take the first field where it finds the character that has been specified, as it searches left to right, top to bottom.

In this example, if the 'W' is not present in a preceding field on the screen, the first occurrence of 'W' will be found in 'NEW TOTAL...'

If the first occurrence of the character does not exist in the field to be referenced, you can enter a unique character in both the field to be referenced and in the PLACEMENT field.

EXAMPLE:

By entering an '\*' in the field to be referenced and in the PLACEMENT field:

```

      8
-----
!
2 !      NE* TOTAL : ZZ9.9      !
!
!
! 0: C1 CH:      ACTION:E LINE NU.:      PLC: * DLM:      !
! ETC.      !
-----

```

The user obtains:

```

      8
-----
!
2 !      NE* TOTAL : ZZ9.9      !
!
!
! 0: C1 CH:      ACTION:M LINE NU.: 100      PLC:      DLM:      !
! ETC.      !
-----

```

At this point, the field can be modified. After the modification is entered, the screen will come back with all modifications in place and the unique character entered will not appear. (i.e. the '\*' will disappear from 'NEW TOTAL').

#### DELIMITER (DLM)

The DLM field is used to enter a fixed label. It can be entered directly on any part of the layout, using a special character to indicate where the label begins and ends.

### EXAMPLE:

```
-----  
!                                     !  
!      *REPORT *                       !  
!                                     !  
! 0:C1 CH:          ACTION:C LINE NU.: 100 PLC: DLM:* !  
! etc.                                     !  
-----
```

The system searches for the character entered in the DLM field ('\*' in the example) in order to create the label 'REPORT' on the Layout. The user would then indicate its positioning, attributes, etc.

It is possible to create a label directly in a specific position on the Layout without using the POSITIONING fields. To do so, the same special character has to be entered in both the PLACEMENT and DELIMITER fields; the POSITIONING fields should NOT be entered. In our example, the positioning is absolute, line 2 and column 8:

```
      8  
-----  
2 !      *REPORT          *             !  
!                                     !  
! 0:C1 CH:          ACTION:C LINE NU.: 100 PLC:* DLM:* !  
! etc.                                     !  
-----
```

### ACTION CODES

The different possible values of the ACTION CODE are listed in the 'Character mode User Interface' guide.

The value 'E', specific to this screen, is used to 'extract' all the information (line number, positioning, attributes, nature, etc.) concerning a data element or label and copy it into the input area at the bottom of the screen. In order to do so, a special character must be entered in the field that you wish to inquire about and also in the PLACEMENT field.

### HOW TO RESOLVE SCREEN LAYOUT ERRORS

When creating a layout, it may happen that some fields are incorrectly positioned or overlap with other fields.

The error is not always easy to locate. If this is the case, it is then advised to use the View of Address of Elements (-ADR) screen, which is accessed with the following CHOICE:

CH: -ADR

This screen gives the absolute positioning of all fields.

The erroneous line is highlighted and preceded by the '>' character, which facilitates corrections.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1	ACTION CODE (REQUIRED)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		C	Creation of the line
		M	Modification of the line
		D	Deletion of the line
		A	Deletion of the line
		T	Transfer of the line
		B	Beginning of multiple deletion
		G	Multiple transfer
		?	Request for HELP documentation
		E or -	Inhibit implicit update
		X	Implicit update without upper/lowercase processing (on certain lines only)
			On the GP-C4 screen (JCL command lines), upper/lowercase processing.
			On the GP-C1 screen, upper/lowercase processing on continuation lines only.
2	3		Line number (REQUIRED)
			PURE NUMERIC FIELD
			It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
3	1		PLACEMENT CHARACTER FOR REFERENCE
			The placement character is used to identify a field on the screen layout.
			This must be either the first occurrence of a given character on the screen, or else, a unique character which does exist on the screen. Special characters can usually be used: '&', '/', '*', etc. (See Paragraph 'How to modify the Screen Description' in this subchapter).
4	1		DELIMITER CHARACTERS OF LITERAL
			The delimiter is a character used to reference the beginning and the end of a fixed label on the screen.
			This must be a unique character which does not exist on the screen. Special characters can usually be used: '&', '/', '*', etc.
			(See Paragraph 'How to modify the Screen Description' in this subchapter).
5	6		DATA ELEMENT OR SCREEN CODE TO CALL
			ELEMENTARY DATA ELEMENT DEFINED IN THE DICTIONARY
			The default is the conversational format defined in the Dictionary. If none has been defined:
			.A Data Element displayed and protected will be in output format.
			.An unprotected Data Element will have an extended internal format, possibly including one or two additional positions for a sign and/or a decimal point.
			GROUP DATA ELEMENT DEFINING A CATEGORY

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Its definition in the dictionary is optional.
		SPECIAL DATA ELEMENTS
	ERMSG	This is used to allocate the placement of error messages. Its default length is 72 characters and it may be repeated vertically.
		It can be modified if ERMSG is defined in the Dictionary with a different format.
		Its NATURE OF THE DATA ELEMENT value must be 'P'.
	PFKEY	A programmable function key (PFkey) may be used:
		.for branching to another screen. The screen to which the dialogue branches and the PF key number are indicated in the update and source fields.
		.to initialize the screen operation code (OPER). The initialization character of the 'OPER' variable and the PFkey number are indicated in the UPDATE and SOURCE fields.
		PFKEY can be positioned anywhere on the '-CE' screen. It is processed in the generated program in:
		.F0520, in order to initialize the operation code of the screen,
		.F20, in order to ensure branching to a subsequent screen. In this case PFKEYs are processed according to the order of their input on the '-CE' screen.
		The value of this function key depends on the generation variant. It is generally numerical and corresponds to the number of the keys available on the keyboard.
		CICS
		The following values may also be used:
		A1 : PA1 key
		A2 : PA2 key
		EN : ENTER key
		00 : CLEAR key
		TANDEM
		Only PFKEY values ranging from '01' to '32', corresponding to keys 'F1' to 'SF16', are taken into account.
		DEC/VAX
		The values used for PFKEY coding are '01', '02', '03', '06' to '11' and '14' to '20'.
		MICROFOCUS, BOS/TP
		Values ranging from '01' to '24' may be used for the PFkeys.
		HP3000
		Values ranging from 1 to 8 may be used for the PFkeys.
	*PASWD	WITH IMS ONLY
		The ability to set a password on a specific screen (PASSWORD macro generation at the MID level).
		This data element must be described in the Specifications Dictionary and its length cannot exceed eight characters.
		IMS : MANAGEMENT OF LOGICAL PAGING

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			(For more details on the management of logical paging, refer to chapter 'Description of a Transaction', subchapter 'Dialogue Complement'.)
			In order to distinguish the different MOD LPAGEs, you must define, for the COND parameter used in the LPAGE macro, a 4-character Element which will receive the last four characters of the screen called.
			This Element must be called in the screen (its NATURE is automatically set to 'P'). Also see the description of the TYPE OF SOURCE FOR DISPLAY and SOURCE FOR DISPLAY/FIRST PART fields.
			Logical paging operator: A 5-character element must be described. Also, see the description of the TYPE OF SOURCE FOR DISPLAY and SOURCE FOR DISPLAY/FIRST PART fields.
			CALL OF ANOTHER SCREEN
			Code of the screen to be called (with value 'S' in the NATURE OF THE DATA ELEMENT field).
			If there is a screen call within the called screen, that call will be ignored.
			EXAMPLE: If screen 'Y' is called into screen 'X', any calls made by screen 'Y' are ignored.
			NAME OF A SCREEN
			If a screen code is used in this field (value 'T' in the NATURE OF THE DATA ELEMENT field), the clear name of the screen will appear. This is often used on a menu-type screen to allow you to select the screen by its title.
			FIXED LABELS
			A fixed label (NATURE OF THE DATA ELEMENT = 'L')
			may be created on the Screen Call of Elements (-CE) screen: the user must first create a line (O: C1) with NO value in this field, and enter the fixed label in the LITERALS field (O: C2).
			The characteristics of this label, apart from its value, can be modified using the 'C1' option of this screen.
			All of the characteristics can be modified on the Screen Layout (-L) or Screen Mapping (-M) screens.
			The 'slash' (/) character serves as a delimiter at the end of the label. It is however possible to include slashes in the label. The last slash is then considered as the delimiter.
			A fixed label is limited to 30 characters, except if it is made up of only one repeated character. In this case, the limit is extended to 160 characters.
6	1		POSITIONING TYPE
		blank	RELATIVE POSITIONING
			The relative starting point is calculated from the end of the previous data element.
			The following examples show, using 'x', the first position of the relative positioning calculation for the following Data Element, with different presentation types and with repetitions of the Data Element.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			LABEL LABEL.....: DATA ELEMENTx
			DATA ELEMENTx
			LABEL LAB: ELEM(1) LAB: ELEM(1)x
			ELEM(1) ELEM(1)x LAB: ELEM(2) LAB: ELEM(2)
			ELEM(2) ELEM(2)
			ELEM(3) ELEM(3) etc.
			For screen calls, the lower right hand corner of the called screen is considered the end position.
		A	ABSOLUTE POSITIONING
			The line and Column numbers correspond to the absolute address of the Data Element on the screen.
			MODIFICATION OF POSITIONING
			Used to modify the positioning with the Screen Layout (-L).
		+	Use '+' to adjust this Data Element's previous location. The number you specify will be added to the original column or line number.
		-	Use '-' to adjust this Data Element's previous location. The number you specify will be subtracted from the original Column or line number.
			NOTE ON POSITION
			For Data Elements with column labels, the starting position is the first character of the field.
			For Data Elements with labels, the starting position is the first character of the label.
			For Data Elements without either, the starting position is the first character of the Data Element itself.
			NOTE: Elements identifying screen group categories have a position. The first Element within the category is therefore positioned relatively to the beginning of this category.
7	2		LINE NUMBER POSITIONING
			PURE NUMERIC FIELD
			IN RELATIVE POSITIONING
			This is the number of lines between two Data Elements. If the presentation option corresponds to a column heading, the number of lines in the column heading is automatically added to the indicated number at the generation level.
			IN ABSOLUTE POSITIONING
			This is the line number on the screen (checked by the System in case of overflow).
			If the line and column numbers are both equal to zero (default values), the Data Element is automatically positioned on the next available tabulation point.
8	3		COLUMN NUMBER POSITIONING
			PURE NUMERIC FIELD
			IN RELATIVE POSITIONING

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		When the line number is zero, a number in this field represents the relative displacement from the previous Data Element on this line, and will be placed on this line.
		When the line number is not zero, the number in this field is the absolute column address.
		IN ABSOLUTE POSITIONING
		This is the address of the Data Element on the line.
		A zero value will correspond by default with column 2 in generation (col. 1 is reserved for the attribute). If the value is 1, the attribute is on the preceding line.
		The system detects line length overflow and field overlapping. However, it does not verify the presence of a space in front of each field.
		If the line and column numbers are both equal to zero (default values), the Data Element is automatically positioned on the next available tabulation point.
9	1	NATURE OF THE DATA ELEMENT
		There are three different types of entities whose natures are defined.
		ELEMENTARY DATA ELEMENTS
	F	Data Element displayed and protected on the screen, but received by the program.
	P	Data Element displayed and protected on the screen, and not received by the program.
		In these two cases, if the Data Element does not have a conversational format, its output format is used.
	V	Data Element displayed, not protected on the screen (user input allowed), and received by the program.
		In this case, if the Data Element does not have a conversational format, its extended internal format is used (including sign and decimal point if relevant).
	O	Name or label of the Data Element only.
	L	Literal.
		The value of the literal is entered in the LITERALS field.
		For complete details, refer to field DATA ELEMENT OR SCREEN CODE TO CALL.
	T	Title of the screen (Clear name).
		DATA ELEMENT DEFINING A SCREEN GROUP CATEGORY
		These Data Elements separate the screen into 3 parts:
		top, repetitive or bottom.
		All Data Elements appearing after a 'group' type Data Element will belong to the category defined for the group, until the next group type Data Element or until the end of the list.
		Data Elements appearing before the first group Data Element belong to the screen top category.
	R	Beginning of the list of entities of the repetitive group.



NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Z	Beginning of the list of entities of the bottom group.
			There can be a maximum of one group Data Element of each type per screen.
			Group Elements may be associated with a positioning value and a label type, which is then the default value for the elements belonging to the group.
			Repetitions may also be defined for the repetitive group.
			CALL OF A SCREEN
		S	Call of another screen
			The code of the called screen is indicated in place of the Data Element code. It will appear in the cross-references.
		W	TANDEM
			Call of an OVERLAY screen. The description of the called screen will appear in 'overlay' in the SCREEN-SECTION. You are responsible for its management.
10	1		LABEL TYPE
			This option allows the choice of the label associated with each Data Element on the screen.
			At the screen level, it is used to override the global selections made at the dialogue level.
			There are four possible labels:
			.The long label (36 positions): Data Element name entered on the Data Element Definition screen,
			.The short label (18 positions), entered on the Data Element Description screen (-D),
			.The relational label (18 positions), entered on the Data Element Description (-D) screen,
			.The column heading label, entered on the Data Element Description screen (-D).
			These labels are entered on the Data Element Description (-D) screen.
		blank	Defaults to the value specified at the Dialogue level. (This option is not authorized for the Dialogue Description level).
		L	Data Element preceded by the long label (Data Element name) left-justified in 36 positions (default value at the dialogue level).
			Example:
			LABEL.....: Data Element
		M	Data Element preceded by the long label (Data Element clear name), right-justified, in 36 positions.
			Example:
			LABEL: Data Element
		N	Data Element preceded by the long label (Data Element name), left-justified with suppression of blanks on the right.
			Example :
			LABEL: Data Element

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	S	Data Element preceded by the short label, left-justified, in 18 positions.
	T	Data Element preceded by the short label, right-justified, in 18 positions.
	U	Data Element preceded by the short label, left-justified, with suppression of blanks on the right.
	R	Data Element preceded by the relational label, left-justified in 18 positions.
		NOTE: For all of the above options, the ':' character is generated (except for labels whose NATURE = 'O').
	F	Data Element only.
	1	Data Element topped by a single line column heading.
		Example:
		HEADING
		Data Element
	2	Data Element topped by a two-line column heading.
	3	Data Element topped by a three-line column heading.
		For the last three options, the longest length of the Data Element and the label is taken into consideration for relative positioning on the screen.
		NOTE: When used at the screen description level, a LABEL TYPE entered for an Element defining a screen category will cause the Data Elements within the category to default to this type.
		ATTRIBUTES OF THE FIELD
		There are three types of field attributes : Intensity, presentation and color.
		In the following 'attribute' fields, the first is for the label or Data Element, the second is for the Data Element.
11	1	INTENSITY ATTRIBUTE - LABEL
	blank	Default to the Dialogue Definition.
	N	Normal intensity.
		(Default value at the Dialogue level).
	B	Bright (Highlighted).
	D	Dark (No display).
12	1	INTENSITY ATTRIBUTE - DATA
	blank	Default to the Dialogue Definition.
	N	Normal intensity.
		(Default value at the Dialogue level).
	B	Bright (Highlighted).
	D	Dark (No display).
13	1	PRESENTATION ATTRIBUTE - LABEL
	blank	Default to the Dialogue Definition.
	N	Normal presentation.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			(Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
14	1		PRESENTATION ATTRIBUTE - DATA
		blank	Default to the Dialogue Definition.
		N	Normal presentation.
			(Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
15	1		COLOR ATTRIBUTE - LABELS
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
16	1		COLOR ATTRIBUTE - DATA
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
17	1		CURSOR DEFAULT POSITION/ SKIP OPTION
			This field has a double role:
		Y	Place the cursor on this Data Element when this screen is initially invoked.
			IMS configuration:
			If this value is not entered, the cursor is positioned on the first Element of the screen; except in the case of LOGICAL PAGING (LPAGE OPTION) where the cursor is positioned at the top of the screen.
18	2		HORIZONTAL REPETITIONS
			PURE NUMERIC FIELD

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Indicates the number of occurrences of a Data Element on the same line. The label occurs only if it is a Column heading. Each occurrence of a Data Element is separated by a space. This repetition is only possible for an elementary Data Element.
		REPETITIVE CATEGORY:
		The number of horizontal repetitions of the first Data Element in the repetitive category helps determine the number of lines for each occurrence.
19	2	VERTICAL REPETITIONS
		PURE NUMERIC FIELD
		For an elementary Data Element:
		The number of vertical occurrences can be combined with the number of horizontal repetitions. The Data Element label also occurs vertically, except if it is a Column heading.
		For a repetitive category:
		All the Data Elements and labels of the repetitive part, which may contain several lines, are repeated vertically. Column headings appear only once.
20	1	PRESENCE VALIDATION OF DATA ELEMENT
		This validation is performed under the condition of the value in the VALIDATIONS CONDITIONS/SET VARIABLES field.
		Omissions to the input field will be taken as low values and converted into spaces prior to validation.
	E	Associated to PFKEY, validates the screen branching values in the generated program.
	blank	Optional Data Element.
	R	Required Data Element. It must be entered when creating a Segment, and cannot be deleted in a modification.
	S	Identical to value 'R', but the '-D' of the Data Element is not used for validations (if it appears on the first call line.)
	P	Optional Data Element. The validation is conditioned by the presence of the Data Element if the value 'P' is on the first call line of the Data Element.
		The update is conditioned by the presence of the Data Element for the Segment indicated on the call line which has the value 'P'. (UPDATE OPTION values: 'M', '+', '-').
	F	Identical to the value 'P', but the '-D' of the Data Element is not used for validations (if it is on the first call line.)
	N	MUST be entered on the Data Element's FIRST call line. This option inhibits the validations entered on the Data Element description lines. Only validations entered on the '-CE' screen are taken into account.
21	1	VALIDATION CONDITIONS/SET VARIABLES
		This field has three different uses:
		1. Data Element used as operation code for the screen.
		2. Data Element used as transaction code for the category.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		3. To specify under which conditions to execute validation and transfers of the Data Element.
		At the Screen level :
		The OLSD Function generates the automatic processing of 6 standard operations - display, update, screen branching, end of conversation, same screen, call of another screen - according to the values of an internal operation code (OPER, see the standard structure of the procedure in the Appendix of the 'On-Line Systems' manual.
	O	Operation code.
		The Data Element is identified as the external operation code. The internal operation code (OPER) will be set to the value placed in the UPDATE TARGET field when the Element takes the value entered in the SOURCE field; the UPDATE OPTION (U) field of the lines on which these values are entered must be set to 'V'.
		NOTE: 'O' may be omitted after the first of a list of operation code values is named.
		The following values may be used in the UPDATE TARGET field:
		'A' : Inquiry
		'M' : Update
		'O' : Transfer to another screen
		'P' : Redisplay
		'S' : Screen continuation (scroll)
		'E' : End
	I	Transaction Code.
		The Data Element is identified as the external transaction code. The internal transaction code (CATM) will be set to the value placed in the UPDATE TARGET field when the Element takes the value entered in the SOURCE field; the UPDATE OPTION (U) field of the lines on which these values are entered must be set to 'V'.
		The following values may be used in the UPDATE TARGET field:
		'C' : Create
		'M' : Modify
		'A' : Delete
		'X' : Create or modify as appropriate
		The Element nature must be 'V'.
	T	Validation condition.
		Validate when creating, modifying or deleting.
	X	Validation condition.
		Validate when creating or modifying.
	A	Validation and update condition.
		Validate under all conditions, but do not update the Segment.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	blank	Validation condition.
		Validate under all conditions.
22	1	UPDATE OPTION
		This field has a two-fold purpose:
		1. With Data Elements that control screen branching in a dialogue:
	S	Simulate screen branching to the screen named in the UPDATE TARGET field as specified by the Operation Code (OPER). Note: this simulation will only occur with options C1 or C3 on the Simulation (-SIM) screen.
	G	Generate (and simulate) the screen branching to the screen named in the UPDATE TARGET field as specified by the Operation Code (OPER).
		NOTE: If there is no screen named in the UPDATE TARGET field, the dialogue is ended.
		The value 'G' may be used with delayed branching (See description of the TYPE OF SOURCE FOR DISPLAY field).
		2. With Data Elements used in update:
	V	Value to validate: Comes as a complement to a Data Element's Description lines in order to indicate additional values to be validated, or to specify the internal values of the Operation Code (OPER) or transaction code (CATM) (with value 'O' or 'I' in the VALIDATION CONDITIONS/ SET VARIABLES field).
		For the Data Element PFKEY being used as the Operation Code (OPER), use the value in the UPDATE TARGET field for the value of the operation code (with value 'O' in the VALIDATION CONDITIONS/SET VARIABLES field).
	M	Update the Segment specified in the UPDATE TARGET field by substitution (MOVE). Default option.
	+	Add the contents of the Element to the Element named in the UPDATE TARGET field.
	-	Subtract the contents of the Element from the Element named in the UPDATE TARGET field, and store the result in the latter field.
		NOTE: Updates are executed only if no errors are found on the screen.
	P	Perform the standard validations on the Data Element, and then perform additional validations as indicated in the sub-function named in the UPDATE TARGET field, using the Element named in the SOURCE FOR DISPLAY field (the Element must be a standard field: the TYPE OF SOURCE FOR DISPLAY field must be left blank).
		NOTE: If there is a class or presence error, the sub-routine will not be performed. However, for Elements that fail the content validation (the valid values of the Data Element as specified on the Data Element Description Screen (-D)), the sub-routine will be performed.
23	1	FIELD CONTENTS IDENTIFIER
	blank	The field contains a fixed label value.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	I	The field contains an initial value automatically displayed when the screen is invoked.
	P	The field contains a presentation value overriding the default value defined at the Data Element level.
	A	This value indicates that the following label is made up of one character repeated more than 30 times.
		INPUT EXAMPLE:
		LABEL
		T LITERALS
		A 045-
		The corresponding label is a line of 45 dashes.
		UPDATE TARGET
		In the normal case, it contains the Segment code and the Data Element (if different from -CE) to be updated.
		A Data element declared as protected in the -CE screen of an O-entity is not moved to the Segment in the F30 function.
		Associated with UPDATE OPTION 'S' or 'G', this group Data Element contains the code of the called screen.
		Associated with UPDATE OPTION 'P', this group Data Element contains the sub-function called.
		With a 'V' UPDATE OPTION:
		. In relation to a Data Element, this field contains the value to validate, or if the value in VALIDATION CONDITIONS/SET OF VARIABLES field is 'O' or 'I', it contains the possible values of OPER or CATM.
		. In relation to the PFKEY Data Element, this field contains the possible values of the internal operation code OPER.
		The possible values of the OPER and CATM variables are described in the Appendix of this manual. It is also possible to indicate these values on the description screen of the specific Data Element.
		For further details, please refer to the 'Data Dictionary' manual.
24	4	UPDATE TARGET / FIRST PART
		In the generated Program associated to the Screen, this field is the Segment code to which belongs the Data Element to be updated from the Data Element in the Screen.
		In the normal case, it contains the Segment code and the Data Element (if different from -CE) to be updated.
		A Data Element declared as protected in the -CE screen of an O-entity is not moved to the Segment in the F30 function.
		Associated with UPDATE OPTION 'S' or 'G', this group Data Element contains the code of the called screen.
		Associated with UPDATE OPTION 'P', this group Data Element contains the sub-function called.
		With a 'V' UPDATE OPTION:

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		. In relation to a Data Element, this field contains the value to validate, or if the value in VALIDATION CONDITIONS/SET OF VARIABLES field is 'O' or 'I', it contains the possible values of CATM or OPER.
		. In relation to the PFKEY Data Element, this field contains the possible values of the internal operation code OPER.
		The possible values of the OPER and CATM variables are described in the Appendix of this manual. It is also possible to indicate these values on the description screen of the specific Data Element.
		For further details, please refer to the 'Data Dictionary' manual, chapter 'Data Elements'.
25	6	UPDATE TARGET / LAST PART
		(Default Option: Data Element code)
		The default option also works for a modification.
26	1	TYPE OF SOURCE FOR DISPLAY
		Specifies the contents of the source field which will be used to fill in a display Data Element.
	blank	The source field is a standard field.
	*	The source field is not a standard field. In this case, you may code any data-name in the source field, which will be taken as such in the generated program.
	D	For Data Elements used as the Operation Code (OPER), when used following a 'G' for generate in the UPDATE OPTION field, causes a delayed branching to the requested screen. The branching will occur after all validations and updating of the reception part of the program have been completed (Functions F05 through F3999-FN). The OPER value is set to 'O'.
	M	Same option as the value 'D' above, except that the OPER value is set to 'M', the OPERD value to 'O'.
	S	IMS:
		Pinpoints the logical paging operator or the field corresponding to the COND parameter in the LPAGE macro.
		SOURCE FOR DISPLAY
		In the normal case, the code of the Segment and of the Data Element source (if different from -CE) are indicated.
		If UPDATE OPTION value = 'S' or 'G', this field contains the value of the field (or the PA key) used to call the requested screen.
		If UPDATE OPTION value = 'V' and with value 'O' or 'I' in the VALIDATION CONDITIONS/SET VARIABLES field, this field contains the value for the initialization of the internal operation code (OPER) or of the internal transaction code (CATM).
		In both cases, such values should be entered without quotes.
		If UPDATE OPTION value = 'P' (PERFORM), this field contains the code of the field used to temporarily store data from the input screen field before the sub-function branching.
27	4	SOURCE FOR DISPLAY / FIRST PART



NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			In the generated program linked to the screen, this field corresponds to the Segment code to which the Data Element updated from the screen Data Element belongs.
		OLP	IMS: Logical paging operator.
		COND	IMS: This value specifies the field corresponding to the COND parameter used in the LPAGE macro.
			In the normal case, the code of the Segment and of the Data Element source (if different from -CE) are indicated.
			If UPDATE OPTION value = 'S' or 'G', this field contains the value of the field (or the PA key) used to call the requested screen.
			If UPDATE OPTION value = 'V' and with value 'O' or 'I' in the VALIDATION CONDITIONS/SET VARIABLES field, this field contains the value for the initialization of the internal operation code (OPER) or of the internal transaction code (CATM).
			In both cases, you do not need to enter the values with quotes, except if the value is blank.
			If UPDATE OPTION value = 'P' (PERFORM), this field contains the code of the field used to temporarily store data from the input screen field before the sub-function branching.
28	6		SOURCE FOR DISPLAY / SECOND PART (Default option: Data Element code)
			If the source Data Element is the same as the Data Element in the screen, this field remains blank.
29	2		LEVEL OF GENERATED SUB-FUNCTIONS PURE NUMERIC FIELD
		06 to 98	The On-line Systems Development function automatically generates the following in the program associated with the screen:
			.A sub-function of Function 20 for each Data Element to be validated,
			.A sub-function of Functions 25, 35 and 60 for each Segment which is accessed in the screen.
			See the 'Structured Code' manual for the definitions of function, sub-function and level notions. Refer to subchapter 'Standard Structure of the Procedure Division' in the Appendix of the 'On-Line Systems Development' manual for the list of functions of the generated program.
			The hierarchical level is 10 by default. It can be modified for a Data Element or a Segment.

## Screen Mapping (-M)

Another screen layout facility is available and offers a new feature: as opposed to the Layout (-L), line numbers need not be specified. The Screen Mapping facility is an on-line tool accessed with the following input in the CHOICE field:

CH: OaaaaaM

Five special characters are needed to implement the Mapping facility. They are described in the following chart. However you may choose other values by overriding the default values at the bottom of the screen before beginning the Mapping.

FUNCTION	SPECIAL CHARACTERS
window beginning delimiter	<
window ending delimiter	>
relative positioning	£
absolute positioning	\$
modification request	?

These special characters allow direct positioning of the fields onto a blank screen whose bottom line displays the set values of the special characters. You can change these characters here according to your convenience.

Once the positioning characters are entered (with or without a sequence order) each field's characteristics are entered in the input fields on the command line at the bottom of the screen.

A CHOICE field (RET:) is also displayed on the left part of this line. It is only when a choice is entered and the ENTER key pressed that mapping transactions are taken into account.

The Mapping (-M) screen may be used to update a screen layout (-L) and its field's characteristics.

>>>> If you use the VisualAge Pacbase WorkStation, the graphical interface of the corresponding windows is described in the VisualAge Pacbase WorkStation Reference manual.

#### EXAMPLE OF USE

The following example illustrates the Mapping (-M) method:

You want three Data Elements displayed on a screen, the first one with absolute positioning and the other two with relative positioning.

```

+-----+
! $           !
! £      £     !
!             !
!             !
!             !
+-----+
-->ENTER<--

```

The system responds with a screen displaying the same characters followed by an order number and command line at the bottom in which you enter information on the first Data Element's characteristics.

```

+-----+
! $1           !
! £2    £3     !
!             !
!-----!
!POS:1 D.E:   (...)!
+-----+

```

You enter the characteristics of the first Data Element coded NUMBER.

```

+-----+
! $1           !
! £2    £3     !
!             !
!-----!
!POS:1 D.E:NUMBER (...)!
+-----+
-->ENTER<--

```

The System displays the first field at its absolute position (NUMBER:....) and asks for the characteristics of the second Data Element (POS:2) on the bottom line.

```

+-----+
! NUMBER:....  !
! £2    £3     !
!             !
!-----!
!POS:2 D.E:NAME (...)!
+-----+
-->ENTER<--

```

The System displays the second field with relative positioning (NAME:....) and asks for the characteristics of the third data element (POS:3). The System will reposition the third field if the initial space provided for the second is not sufficient.

```

+-----+
! NUMBER:....  !
! NAME:.... £3  !
!             !
!-----!
!POS:3 D.E:CITY (...)!
+-----+
-->ENTER<--

```

You enter the characteristics of the third Data Element. The system displays the completed screen layout.

```

+-----+
! NUMBER:....  !
! NAME:.... CITY:.... !
!             !
!             !
!             !
+-----+

```

### ADDITIONAL FIELDS

In order to add new fields on a screen, the same Mapping (-M) method applies.

#### EXAMPLE

You want to add a fourth field on the screen described in the preceding example.

You enter the special character which correspond to the positioning you want.

```

+-----+
! NUMBER:.... !
! NAME:.... CITY:.... !
! £ !
! !
! !
+-----+
-->ENTER<--

```

The System automatically assigns an order number (POS:1), and requests the characteristics of the field.

```

+-----+
! NUMBER:.... !
! NAME:.... CITY:.... !
! £1 !
!-----!
!POS:1 D.E:BIRTH (...)!
+-----+

```

You enter the characteristics of the fourth field, the Data Element coded BIRTH.

```

-->ENTER<--
+-----+
! NUMBER:.... !
! NAME:.... CITY:.... !
! BIRTHDATE:..... !
! !
! !
+-----+

```

The System displays the completed screen layout.

### HOW TO CHANGE THE SCREEN LAYOUT

If you want to change the positioning of some fields, these must be identified. This is done by using the window's beginning and ending delimiters: < and >.

```

+-----+
! NUMBER:.... !
! <NAME:.... CITY:....>!
! BIRTHDATE:..... !
! !
! !
+-----+
-->ENTER<--

```

The System displays the fields in the window with their respective order number and with their initial positioning.

Here, £1 and £2 for NAME and CITY.

```

+-----+
! NUMBER:.... !
! £1.....£2..... !
! BIRTHDATE:..... !
! !
! !
+-----+

```

You can delete existing fields, create new ones or change the existing layout. In this example, the £2 position is to be deleted, £1 is to be moved and a new field is to be created (£).

```

+-----+
! NUMBER:.... !
! £      £1   !
! BIRTHDATE:..... !
!           !
!           !
+-----+

-->ENTER<--

```

The System automatically assigns a new order number (£3) and displays the information request line (POS:3).

```

+-----+
! NUMBER:.... !
! £3      NAME:.... !
! BIRTHDATE:..... !
!-----!
!POS:3 D.E:TYPE (...)!
+-----+

```

You enter the characteristics of the field TYPE.

```

-->ENTER<--

+-----+
! NUMBER:.... !
! TYPE :.... NAME.... !
! BIRTHDATE:..... !
!           !
!           !
+-----+

```

The System displays the completed screen layout.

### HOW TO GET INFORMATION ON A FIELD AND UPDATE IT

When you need to check the code and characteristics of a given field, you must identify this field on the screen with the window beginning and ending characters. The System returns a screen with the window containing the '£' or '\$' characters followed by order numbers. You enter '?' on the special character identifying the desired Data Element.

The System responds with a screen specifying the data element code and its characteristics on a command line at the bottom of the screen.

You may update the Data Element's characteristics on this line.

#### EXAMPLE

If you want to check the code and characteristics of the second Data Element enters '<' and '>'.

```

+-----+
! NUMBER:.... !
! <NAME>.... CITY:.... !
! BIRTHDATE:..... !
!           !
!           !
+-----+

-->ENTER<--

```

The System displays the Data Element positioning character with an order number (here the order number is 1).

```
+-----+
! NUMBER:.... !
! £1      CITY:.... !
! BIRTHDATE:..... !
!          !
!          !
+-----+
```

You enter a '?' for £1, the '?' also being a positioning character.

```
+-----+
! NUMBER:.... !
! ?1      CITY:.... !
! BIRTHDATE:..... !
!          !
!          !
+-----+
-->ENTER<--
```

The System returns the same screen and the bottom line contains the Data Element code and characteristics which you may update.

```
+-----+
! NUMBER:.... !
! ?1      CITY:.... !
! BIRTHDATE:..... !
!-----!
!POS:1 D.E:NAME (...)!
+-----+
```

### REPETITIVE CATEGORY LAYOUT METHOD

In order to create a repetitive category on a screen you should position £R or £R where this category starts. Then Data Elements belonging to this category are positioned using the Mapping (-M) method described above.

It is possible to define a repetitive category made up of several lines per occurrence.

The number of lines per occurrence is specified by the number of horizontal repetitions of the first data element in the repetitive category.

### EXAMPLE

The repetitive category is:

```
DATE:..... QUANTITY ORDERED:.....
```

If '00' is the number of horizontal repetitions of the first Data Element and '03' the number of vertical repetitions, the System displays the following layout:

```
DATE:..... QUANTITY ORDERED:.....
DATE:..... QUANTITY ORDERED:.....
DATE:..... QUANTITY ORDERED:.....
```

If '02' is the number of horizontal repetitions of the first Data Element, the System displays the following layout:

```
DATE:..... QUANTITY ORDERED:.....
DATE:..... QUANTITY ORDERED:.....
DATE:..... QUANTITY ORDERED:.....
```

In this case, the repetitive category is made up of two lines per occurrence.

### NOTE

When the height of a repetitive category (i.e., the number of lines per occurrence) is modified, for example due to the addition of a new Data Element, the number of horizontal repetitions is updated automatically by the System.

This is only valid on the Mapping (-M) screen.

For the screen-bottom, enter \$Z or £Z.

### EXAMPLE

You enter \$R -- specifying a repetitive category -- and the positioning characters corresponding to its Data Elements:

```
+-----+
!$R££      !
!          !
!          !
!          !
+-----+
-->ENTER<--
```

Key in the characteristics of Data Element coded BEGIN:

```
+-----+
!$R£1£2    !
!          !
!          !
!-----!
!POS:R C ELEM. N RH RV !
!RET: R BEGIN R 01 03 !
+-----+
-->ENTER<--
```

Key in the characteristics of Data Element positioned £1:

```
+-----+
! £1£2     !
!  1 2    !
!  1 2    !
!-----!
!POS:1 C ELEM. N RH RV !
!RET: R NAME V      !
+-----+
-->ENTER<--
```

Key in the characteristics of Data Element positioned £2:

```
+-----+
! NAME:.....£2    !
! NAME:..... 2    !
! NAME:..... 2    !
!-----!
!POS:2 C ELEM. N RH RV !
!RET: R AGE V      !
+-----+
-->ENTER<--
```

Display of completed layout:

```
+-----+
! NAME:..... AGE  !
! NAME:..... AGE  !
! NAME:..... AGE  !
```

```

!           !
!           !
!           !
+-----+

```

### RETURNING TO THE INITIAL SCREEN

When entering the Data Elements' characteristics, you may want to return to the initial screen. You can do so by entering the value 'R' in the 'RET:' field.

### USE CONSTRAINTS

The Mapping (-M) screen can only be used with screens for which there are 100 or fewer Data Element call lines (continuation lines included).

No more than 36 Data Elements can be entered at once. As a result, it is not possible to open more than 36 windows at the same time.

It is not possible to call and/or modify Data Elements in column 1. However, this may be done on other screens: Call of Elements (-CE), and the Layout (-L).

### NESTED CATEGORIES

Some screens may require the use of nested categories, for instance, header fields displayed at the top of a repetitive category. In particular, nested categories are used when each line of a repetitive category must be preceded or followed by a specific label.

In order to describe a screen using nested categories with the Mapping (-M) screen, you should keep in mind that when a Data Element is created, it is assigned a category depending on its location on the screen.

This category may be changed in the SCREEN CATEGORY / DATA ELEMENT field in the command line of the Mapping (-M) screen (this field is identified by the fixed label 'C').

#### EXAMPLE

The following screen is to be described:

```

+-----+
!1.NAME:XX   FIRST NAME:XX   !
!2.NAME:     FIRST NAME:     !
!3.NAME:     FIRST NAME:     !
!           !
!           !
+-----+

```

Several description methods are possible:

#### 1. Step-by-step description (by category)

The labels must be defined first, by default they automatically belong to the screen-top category.

```

+-----+
!£1         !
!£2         !
!£3         !
!           !
!           !
+-----+

```

Once the labels are described, the repetitive category must be defined:



```

+-----+
!1.$R£1      £2      !
!2.          !
!3.          !
!           !
!           !
+-----+

```

2. Use of the screen-bottom category

```

+-----+
!£1$R£2      £3      $Z!
!£4          !
!£5          !
!           !
!           !
+-----+

```

The number 1 position corresponding to the '1.' label is automatically assigned to the screen-top category.

Positions 4 and 5 (the '2.' and '3.' labels) belong to the screen bottom category.

3. Override of the SCREEN CATEGORY / DATA ELEMENT field:

```

+-----+
!£1$R£2      £3      !
!£4          !
!£5          !
!           !
!           !
+-----+

```

In this case, positions 4 and 5 belong to the repetitive category whose height (value in the HR field) was automatically set to 3.

The value in the VR field must be set to zero. Otherwise this set of three lines would be repeated.

When positions 4 and 5 are described, the SCREEN CATEGORY / DATA ELEMENT field should be overridden with a blank.

Then the repetitive category must be redefined (enter '?' in front of the 'R' character to access the command line): its height (HR field value) should be set to zero and the number of repetitions should be specified.

#### NOTE ON DELETION

Special attention should be given to deletions when using the Mapping (-M) screen:

When the numbered position of a Data Element is deleted from the Mapping screen, the data element itself is deleted.

EXAMPLE:

```

1. NAME: XXXXXXXX
2.<NAME: XXXXXXXX>
   ---> ENTER
3. £1... .....
4. ££
   ---> ENTER
5. £2£3

```

In this example, the position number '1' corresponding to an existing Data Element is overridden by a positioning character (£). The £1 position is no longer displayed: the Data Element has been deleted.

The two positioning characters (££) indicate that two new Data Elements are to be created and are therefore automatically numbered in sequence (£2£3).

#### DESCRIPTION OF REPETITIVE AND SCREEN-BOTTOM CATEGORIES

When the positions of a category and a Data Element coincide on a screen, only the Data Element's position is displayed.

In order to modify the description of a category, the Data Element's position must be moved in order to make room for the category's position.

Therefore it is recommended to avoid this situation when the description of the category is still tentative.

#### MANAGEMENT OF FIELD OVERLAP AND/OR OVERFLOW

The Mapping (-M) screen includes a facility that prevents field overlap/overflow by moving the field(s) to the right or, if needed, to the following line.

When a field overlap/overflow is not resolved by this facility, erroneous fields are displayed on the screen as numbered positions so that they can be displaced on-line by the user.

#### NOTES

1. When a screen already described (via the Call of Elements (-CE) for example) is called for the first time on the Mapping screen, this correction facility is inhibited so that you can check the layout beforehand.
2. With a repetitive category, the automatic repositioning of a field may change this category's characteristics. For example, when a Data Element is moved down to the next line the implication is that the category's height may be increased (value in HR field of the Data Element in this category). Therefore, we recommend to carefully assign the position of each Data Element when describing it. This will avoid the consequences of automatic repositioning, e.g. overlap on the screen-bottom.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1	RETURN TO INITIAL SCREEN
	R	Return to the initial screen.
2	1	SCREEN CATEGORY / DATA ELEMENT (REQUIRED)
		The value of this Data Element is automatically assigned. However, it can be overridden.
	blank	Data Element belonging to a screen-top category.
	R	Data Element belonging to a repetitive category.
	Z	Data Element belonging to a screen-bottom category.
3	6	DATA ELEMENT OR SCREEN CODE (REQUIRED)
		ELEMENTARY ELEMENT DEFINED IN THE DATA DICTIONARY

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Unless otherwise specified (i.e. no declared conversational format):
		.A Data Element displayed and protected will be in output format.
		.An unprotected Data Element will have an extended internal format, possibly including one or two additional positions for a sign and/or a decimal point.
		GROUP DATA ELEMENT DEFINING A CATEGORY Need not be defined in the dictionary.
		SPECIAL DATA ELEMENTS
	ERMSG	This is used to allocate the placement of error messages. Its default length is 72 characters and it may be repeated vertically.
		It can be modified if ERMSG is defined in the Dictionary with a different format.
		Its nature in the screen must be 'P'.
		CALL OF ANOTHER SCREEN
		If screen 'Y' is called into screen 'X', any calls made by screen 'Y' are ignored.
		NAME OF A SCREEN
		If a screen code is used in this field, the name of the screen will appear. This is often used in a menu type screen to allow the selection of the screen by its title.
		FIXED LABELS
		This field should be left blank when creating literals which can only be defined with operation code 'C2'.
		Its characteristics other than 'value' can be modified on the screen obtained by operation code 'C1'.
		Positioning may be modified on the 'Screen Layout' screens (-L or -M).
4	1	NATURE OF THE DATA ELEMENT (REQUIRED)
		There are three different types of entities whose natures are defined.
		ELEMENTARY DATA ELEMENTS
	F	Data Element displayed and protected on the screen, but received by the program.
	P	Data Element displayed and protected on the screen, and not received by the program.
		In these two cases, if the Data Element does not have a conversational format, its output format is used.
	V	Data Element displayed, not protected on the screen (user input allowed), and received by the program.
		In this case, if the Data Element does not have a conversational format, its extended internal format is used (including sign and decimal point if relevant).
	O	Name or label of the Data Element only.
	L	Literal.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The value of the literal is entered in the LITERALS field.
		For complete details, refer to field DATA ELEMENT OR SCREEN CODE TO CALL.
	T	Title of the screen (Clear name).
		DATA ELEMENT DEFINING A SCREEN GROUP CATEGORY
		These Data Elements separate the screen into 3 parts:
		top, repetitive or bottom.
		All Data Elements appearing after a 'group' type Data Element will belong to the category defined for the group, until the next group type Data Element or until the end of the list.
		Data Elements appearing before the first group Data Element belong to the screen top category.
	R	Beginning of the list of entities of the repetitive group.
	Z	Beginning of the list of entities of the bottom group.
		There can be a maximum of one group Data Element of each type per screen.
		Group Elements may be associated with a positioning value and a label type, which is then the default value for the elements belonging to the group.
		Repetitions may also be defined for the repetitive group.
		CALL OF A SCREEN
	S	Call of another screen
		The code of the called screen is indicated in place of the Data Element code. It will appear in the cross-references.
	W	TANDEM
		Call of an OVERLAY screen. The description of the called screen will appear in 'overlay' in the SCREEN-SECTION. You are responsible for its management.
5	1	LABEL TYPE
		This option allows the choice of the label associated with each Data Element on the screen.
		At the screen level, it is used to override the global selections made at the dialogue level.
		There are four possible labels:
		.The long label (36 positions): Data Element name entered on the Data Element Definition screen,
		.The short label (18 positions), entered on the Data Element Description screen (-D),
		.The relational label (18 positions), entered on the Data Element Description (-D) screen,
		.The column heading label, entered on the Data Element Description screen (-D).
		These labels are entered on the Data Element Description (-D) screen.
	blank	Defaults to the value specified at the Dialogue level. (This option is not authorized for the Dialogue Description level).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	L	Data Element preceded by the long label (Data Element name) left-justified in 36 positions (default value at the dialogue level).
		Example:
		LABEL.....: Data Element
	M	Data Element preceded by the long label (Data Element clear name), right-justified, in 36 positions.
		Example:
		LABEL: Data Element
	N	Data Element preceded by the long label (Data Element name), left-justified with suppression of blanks on the right.
		Example :
		LABEL: Data Element
	S	Data Element preceded by the short label, left-justified, in 18 positions.
	T	Data Element preceded by the short label, right-justified, in 18 positions.
	U	Data Element preceded by the short label, left-justified, with suppression of blanks on the right.
	R	Data Element preceded by the relational label, left-justified in 18 positions.
		NOTE: For all of the above options, the ':' character is generated (except for labels whose NATURE = 'O').
	F	Data Element only.
	1	Data Element topped by a single line column heading.
		Example:
		HEADING
		Data Element
	2	Data Element topped by a two-line column heading.
	3	Data Element topped by a three-line column heading.
		For the last three options, the longest length of the Data Element and the label is taken into consideration for relative positioning on the screen.
		NOTE: When used at the screen description level, a LABEL TYPE entered for an Element defining a screen category will cause the Data Elements within the category to default to this type.
6	1	INTENSITY ATTRIBUTE - LABEL
	blank	Default to the Dialogue Definition.
	N	Normal intensity.
		(Default value at the Dialogue level).
	B	Bright (Highlighted).
	D	Dark (No display).
7	1	INTENSITY ATTRIBUTE - DATA
	blank	Default to the Dialogue Definition.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		N	Normal intensity.
			(Default value at the Dialogue level).
		B	Bright (Highlighted).
		D	Dark (No display).
8	1		PRESENTATION ATTRIBUTE - LABEL
		blank	Default to the Dialogue Definition.
		N	Normal presentation.
			(Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
9	1		PRESENTATION ATTRIBUTE - DATA
		blank	Default to the Dialogue Definition.
		N	Normal presentation.
			(Default value at the Dialogue level).
		B	Blinking.
		R	Reverse video.
		U	Underlined.
10	1		COLOR ATTRIBUTE - LABELS
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
11	1		COLOR ATTRIBUTE - DATA
		blank	Default to the Dialogue selection.
		W	Normal (Default value at the Dialogue level).
		R	Red.
		P	Pink.
		Y	Yellow.
		G	Green.
		T	Turquoise.
		B	Blue.
12	1		CURSOR DEFAULT POSITION/ SKIP OPTION
			This field has a double role:
		Y	Place the cursor on this Data Element when this screen is initially invoked.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		IMS configuration:
		If this value is not entered, the cursor is positioned on the first Element of the screen; except in the case of LOGICAL PAGING (LPAGE OPTION) where the cursor is positioned at the top of the screen.
13	2	HORIZONTAL REPETITIONS
		PURE NUMERIC FIELD
		Indicates the number of occurrences of a Data Element on the same line. The label occurs only if it is a Column heading. Each occurrence of a Data Element is separated by a space. This repetition is only possible for an elementary Data Element.
		REPETITIVE CATEGORY:
		The number of horizontal repetitions of the first Data Element in the repetitive category helps determine the number of lines for each occurrence.
14	2	VERTICAL REPETITIONS
		PURE NUMERIC FIELD
		For an elementary Data Element:
		The number of vertical occurrences can be combined with the number of horizontal repetitions. The Data Element label also occurs vertically, except if it is a Column heading.
		For a repetitive category:
		All the Data Elements and labels of the repetitive part, which may contain several lines, are repeated vertically. Column headings appear only once.
15	1	FIELD CONTENTS IDENTIFIER
		blank
		The field contains a fixed label value.
		I
		The field contains an initial value automatically displayed when the screen is invoked.
		P
		The field contains a presentation value overriding the default value defined at the Data Element level.
		A
		This value indicates that the following label is made up of one character repeated more than 30 times.
		INPUT EXAMPLE:
		LABEL
		T LITERALS
		A 045-
		The corresponding label is a line of 45 dashes.
16	30	DISPLAYED LITERAL
		This field contains either a literal, or an initial or presentation value. If no TYPE OF LITERAL value is entered in the preceding field, this field contains a fixed value. (see field 'DATA ELEMENT OR SCREEN CODE TO CALL')

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		NOTE: The simple or double quote indicated in this area is replaced by a blank if the same quote is indicated as delimiter on the definition screen of the library. This replacement is performed in order to avoid COBOL compilation errors due to the presence of this delimiter in the 'values'.
		IBM 36, IBM 38/AS 400:
	001-099	This value is the INDICATOR number for attribute positioning on the erroneous Data Element. It is entered on the first call line of the Data Element.
		On CONTINUATION lines:
		Additional INDICATORS can be positioned. They are automatically generated in the screen's DDS and managed by Procedural Code (-P) lines. They are formatted as follows:
		IIIOTLD IIIOTLD IIIOTLD
		where:
		III= INDICATOR number (the maximum number of indicators is 99). With IBM 38 and AS/400: the value 'NUM' is used to generate numeric or signed numeric data ele.
		O = IBM 38: ON/OFF INDICATOR switch (N=OFF, SPACE=ON) IBM 36: This position must be left blank.
		T = ATTRIBUTE TYPE
		L = ATTRIBUTE FOR DATA ELEMENT LITERAL
		D = ATTRIBUTE FOR DATA ELEMENT CONTENTS
		With an indicator number:
		ATTRIBUTE TYPE CORRESPONDING ATTRIBUTE (L or D)
		I: INTENSITY
		B: BRIGHT
		D: DARK (i.e. NO DISPLAY)
		P: PRESENTATION
		B: BLINK
		R: REVERSE VIDEO
		U: UNDERLINED
		C: COLOR
		W: WHITE
		G: GREEN
		R: RED
		P: PINK
		Y: YELLOW
		T: TURQUOISE
		B: BLUE
		R: PROTECTION
		P: PROTECTED CONTENTS (in position D)
		If 'NUM' is coded:



NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		O = '*'
		T = 'S' or 'Y'
		If T='S', the Data Element is processed as a 'SIGNED NUMERIC' field.
		If T='Y', the Data Element is processed as a 'NUMERIC ONLY' field.



---

## Chapter 4. Dialogue Simulation (-SIM)

### IMPLEMENTATION

Once the screen layout has been entered, it may be necessary to check the position of entry fields and headers, as well as their characteristics (protected or unprotected fields, attributes, cursor position, fixed or variable, etc.).

This is possible via the Dialogue Simulation facility, accessed with the following CHOICE:

CH: -SIM

With this facility, you can view a SIMULATION of a layout as it would appear when the transaction is actually executed.

This Choice is associated with one of the following Operation Codes, depending on screen size and attributes:

- OPTION C1 - Simulation of the screen layout without color attributes, up to Line 24, Column 80.
- OPTION C3 - Simulation of the screen layout with color attributes and reverse video, up to Line 24, Column 80.
- OPTION C4 - Simulation of the screen layout without color attributes, 43-80 or 27-132.
- OPTION C5 - Simulation of the screen layout with color attributes and reverse video, 43-80 or 27-132.

**NOTE:** The 27-132 layout is simulated when the value in the SCREEN SIZE - NUMBER OF COLUMNS field on the Dialogue/Screen Definition screen is greater than 80. Otherwise, the 43-80 layout is simulated.

### POSITION IN THE LAYOUT

Depending upon the consultation operation executed, the position of a field on the layout will differ. This is due to the position taken by the attribute in simulation ('-SIM').

A field defined in column 'n' will appear in 'n+1' on the '-L' and in 'n' for the '-SIM'.

### NOTE

It is the position of the Data Element which is provided, not the position of the attribute.

Fixed labels and Data Element names are displayed on each line of a repetitive category.

### EXAMPLES

- CICS: In the generated BMS map, the position is recalculated; for one Data Element defined in absolute positioning on line 20, column 50, the generated map contains: DFHMDF POS=(20,49),...

- IMS: For the same Data Element, the generated MID/MOD will contain: DFLD POS=(20,50),...
- BULL: For the same Data Element, POS=(20,50).  
For information regarding message length, please refer to the Operations manual.

### HARDWARE SUPPORTING THE USE OF FUNCTION KEYS

Upon exiting the simulation screen, by pressing the CLEAR key, the screen description screen (-CE) is displayed.

If this key has another significance for the system, it is possible to define an 'EXIT' PFkey by entering the following CHOICE:

CH: -SIM nn

where nn = the associated PFkey number.

In this case, return to the -CE screen is executed by pressing the corresponding PFkey.

### NOTES

The CLEAR key is synonymous with the PFKEY value '00' (except with DEC and IMS).

If a simulation of a layout is requested, and an error occurs, the system will display the View of Address of Elements (-ADR) screen of the given screen. In order to facilitate error correction, the corresponding line is highlighted and preceded by a '>' character.

The PFKEY 'nn' number (used for return to the Definition screen from a '-SIM') must be between 01 and 24. Therefore, it is important not to use PFKEY numbers between 13 and 24 if they are not available on the keyboard. In the latter case, an abnormal end would occur.

For CICS, values A1 for 'PA1', A2 for 'PA2' and EN for ENTER are also possible output values.

With IMS, EN is a possible value.

With this system you can simulate the branching of the different screens that make up the Dialogue.

Two types of operations are necessary:

- Prepare screen branching using the Call of Elements (-CE) screen,
- Ensure the actual branching.

### PREPARATION OF SCREEN BRANCHING

The preparation of screen branching is done on the screen description (on the Call of Elements (-CE) screen or the Screen Layout (-L)). It consists of:

- Defining the Data Element(s) used for screen branching. For hardware using PF keys, the PFKEY Data Element, which is a reserved field, may be used for this purpose.

- Defining whether screen branching will be executed in simulation only (UPDATE OPTION 'S') or also in generation (UPDATE OPTION 'G').
- Defining the screen that it is required to branch to, using the UPDATE TARGET field.
- Defining the value of the Data Element, or the PFKEY number, used for screen branching, in the SOURCE FOR DISPLAY FIELD. i.e. Enter which PFKEY or what value is to be entered into the choice type field to initiate the branching.
- Defining whether the screen will be called before or after reception processing.

If the screen to be called is not entered, the conversation will be terminated in generation and in simulation, and the system will return to the Call of Elements screen.

#### **NOTE:**

For CICS: 'A1' for 'PA1', 'A2' for 'PA2', are taken into account in screen branching.

For other hardware: 'EN' for ENTER key is taken into account in screen branching.

#### THE ACTUAL BRANCHING

On the Simulation (-SIM) screen, you only need to enter the branching value or use the PFkey, as defined on the Call of Elements screen (-CE), to obtain the desired screen branch. If the branching Data Element is included in this next screen, this process can be repeated.

#### NOTES

It is possible to use several fields for screen branching.

If an invalid value for the screen branching Data Element (or PFKEY) is entered, you are presented with the current screen re-displayed.

When screen branching is standardized, it is possible to group this description into a screen, this can be then be called by all the screens of a dialogue.

#### SIMULATION EXIT

This is executed in the same way as for the Screen Layout, i.e. using the CLEAR key, or the PFKEY as defined when you initiate the simulation, if the hardware in use supports this function.

The System displays the -CE screen of the given screen.

In the case of an error in the Screen Layout, the system will branch to the View of Address of Elements (-ADR) screen of the given screen. In order to facilitate error correction, the line(s) in error are highlighted by the placement of a '>' character on that line.



---

## Chapter 5. Description of a Transaction

---

### Introduction

Once the screen has been described, all the Segments necessary to complete its description must be associated with it.

All technical characteristics and the uses of each Segment have to be specified. See subchapter 'Screen Call of Segments'.

The Dialogue Complement includes information common to all of the screens of a dialogue, and data regarding the physical environment of the transaction (See subchapter 'Dialogue Complement').

Certain generated external parameters (file names or modules being used, length of messages, value of attributes, etc.) being directly linked to the hardware in use can be modified on the 'Generation Elements' screen (-GG). See subchapter 'Overriding Generated Constants'.

All additional procedures associated with the screen, which cannot be produced automatically either from the description of the screen or of its Segments, are entered via the Structured Code function:

- Macro-Structures,
- Work Areas,
- Specific procedures.

These entities are not described in this manual. Refer to the 'Structured Code' manual.

---

### Dialogue Complement (-O)

This screen is used to specify the physical characteristics of the Error Message file and all the additional information needed for the generation of an on-line program.

The physical characteristics of the Error Message file are used to generate its description and accesses. The physical accesses are generated in the F8098 Function. If they need to be re-written, they will have to be overridden in this function.

The physical characteristics of the Error Message file, the common area, etc., can be entered regardless of the type of TP monitor in use. Same external name as dialogue error message PCB.

Other options are selected according to the hardware and TP monitor in use at the site.

Different values can be entered in the OPTIONS field where they should be separated by a 'blank' character.

When the screen is generated, a WARNING type message points out erroneous parameters, if any.

## PREREQUISITE

The dialogue must have been previously defined.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	2	DIALOGUE CODE (REQUIRED)
		To define a Dialogue, only these two characters are required.
		These are the first two characters of the complete code.
2	2	COMMON AREA - DATA STRUCTURE CODE
		This is the code of the common area. It is a Data Structure which stores the information that is common to all of the screens of a dialogue. It also stores the necessary Segment access keys.
		As a Data Structure, it may consist of more than one Segment. When this occurs, the Segments will be redefined in the LINKAGE SECTION (WORKING-STORAGE SECTION if appropriate).
		PACBENCH C/S : To be entered in the TUI Client only (C/S Screen entity).
3	1	ORGANIZATION OF ERROR MESSAGE FILE
	V	Indexed Organization (VSAM).
	D	Database Segment (DL/1).
		The description of the error message file and accesses to it are generated automatically.
		In IMS, the error message file description is not generated automatically. You must generate it in a Segment called by a hierarchical block.
	U	Access managed by specific procedural code.
	Y	UNISYS 2200: Indexed organization (SFS) taken into account in the dialogue generation.
		The Y value mentioned in the organization of a Segment called in the -CS of a screen (or in the Dialogue complement for an error message file) will cause:
		- the declaration of Segments in FD
		- the access generation in function 80.
4	8	EXTERNAL NAME OF ERROR MESSAGE FILE
		IBM CICS - VSAM : DDNAME (8 characters)
		DL/1 : mnemonic code of PCB. This code must be in the list of PCB's for the PSB of this transaction.
		DPS7-UFAS: File-code (2 characters).
		DPS8-TP8: External name generated in the SELECT clause (first 4 characters); blocking factor (last 4 char.) (generated in the FD clause).
		OTHER VARIANTS: Logical name, generated in the SELECT clause.
5	4	FIRST SCREEN CODE OF THE DIALOGUE
		In CICS:



NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		If this code is not entered, the Common Area of each screen of the dialogue is generated in the WORKING- STORAGE SECTION.
		If the code is entered, the Common Area of each screen of the dialogue is generated in the LINKAGE SECTION, except for the first screen, where it is generated in the WORKING-STORAGE Section.
		In IMS:
		The system generates a "monitor" program, which calls the indicated screen at the beginning of a dialogue and calls it again at the end of the dialogue (the external names of the first displayed screen and of its corresponding program must have been previously defined at the screen definition level).
		The code is required, even when the MONITOFF option is on.
		In GCOS7 and GCOS8:
		This field is only used with value 'C' for TYPE OF MAP TO GENERATE (Multi-terminal type transaction).
		In DEC/VAX, ICL, MICROFOCUS, HP3000, and BOS/TP:
		The DIALOGUE function generates a branching monitor. This monitor branches to the indicated screen at the beginning of the conversation.
		In UNISYS-SERIES A:
		This field is used for the multi-terminal type transactions (value 'C' for TYPE OF TP MONITOR, MAP TO GENERATE).
		In UNISYS-2200:
		This field is not used.
6	4	NUMER.
		COMPLEMENTARY COMMON AREA LENGTH
		PURE NUMERIC FIELD
		The length of the conversation area is determined by the combined lengths of: the screen code (6 char.), a common area data structure, a field used to store the access keys of displayed segments, and technical information (transparent).
		The length of the memorization field of the access keys is set to 100 characters by default.
		If it exceeds 100 characters, this default option must be modified accordingly.
		Each displayed Segment's access key length is calculated by category.
		The total length of the conversation area is specified in the generated program as the value of the field 5-xxnn-LENGTH.
		EXAMPLE: If a Segment is used in the screen-top category, the repetitive category and the screen- bottom category, the length of the keys may be calculated as follows:
		the length of the key for the screen-top category,
		+ 2 times the length of the key for the repetitive category,
		+ the length of the key for the screen-bottom category.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		For more details, refer to 'The 'On-Line Systems' manual specific to your environment, chapter 'The Generated Program'.
		When the generation option = 'MDT-OFF' or 'DYNPRT' for the BULL multi-screen variant and UNISYS series A (OPTION field), to calculate the key length, the length of the longest screen Segment of the Dialogue must be taken into account.
7	6	CODE OF PSB OR SUB-SCHEMA
		Reserved for Database Segments (DL/1).
		The PCB name of the error message file (external name) must belong to the PSB.
		HP3000: for a monitor generation with Database connection, specify the ALLBASE/SQL Database Block code.
8	47	OPTIONS
		Enter here the options, separating them with blanks. A synonym is associated to some options to allow for the input of all necessary options.
	ATTRE	UNISYS 2200: Definition of default attributes at the screen level.
	ATTRIB	UNISYS 2200: Attributes specified dynamically remain displayed (REVERSE VIDEO and BLINK). This option is positioned in F7020.
	CSSA	Memorizes all SSA's during a generated access to Databases. This option is valid with all the screens of a Dialogue.
		When this option is on, all the screens in the Dialogue must be recompiled because the length of the COMMUNICATION AREA is increased.
	CVISION	GCOS6:
		This option allows to download the standard validations (presence validation, numeric validation, formatting of the initialization character) onto the CVISION screen manager, thus improving performances.
	DCPROC	Generates the DECLARE CURSOR of DB2, NONSTOP SQL and DATACOM tables in DIVISION PROCEDURE, FOC function. If this option is not specified, the DECLARE CURSOR is generated in the WORKING STORAGE SECTION.
	DFHAID	IBM CICS: its contents initializes the PFKEYS Table.
	DYNAM	Pactables table DYNAM-called.
	DYNPRT	Specifies the recognition of the variable field protection in the DE-AT table (position 5) and the automatic processing of the protection in F7020.
		Enter an F in position 5 of the DE-AT table for the given Data Element. The Data Element is then protected and received by the program (nature F).
		This option is not available with the IBM 36, UNISYS series A variant (SDF format).
	F10	Processes any category (top, bottom, repetitive) provided that it contains at least one variable field or access to a Segment.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	FOR	Formatting of generated COBOL. Left-justified DATA DIVISION, no justification for the PROCEDURE DIVISION but one COBOL command per line.
		This option may be entered on the Library Definition screen. In this case, the Dialogue is generated in formatted COBOL even if the FOR option is not explicitly entered in this field.
		WARNING: This option still exists but is no longer maintained.
	HEXVAL	UNISYS A: Generation of fields with HEXADECIMAL value in PIC XX.
	HLPI	Generation of file access with HLPI interface.
	HLPVAL	Sending of values (indicated in -D of a Data Element) to be checked to the error messages file even though this check has been removed from the screen in which the Data Element is called.
	HPRT	Specifies calls to the documentation associated with protected fields (P or F type). This includes the documentation associated with the Data Element (E-D) as well as texts and comments associated in the screen's comments lines (-GC screen, F-type lines
		In the Help program, the screen flow will be: first the help on variable Data Elements, then the help on protected Data Elements (in the order of the -CE screens).
	INDEXS	DL/1: Automatic generation of Segment Accesses via secondary indexes (see subchapter 'Secondary Index Management').
		The INDEXS option is not compatible with the NOP58 option if the target Segment code is different from the source Segment code and if the source Segment is not generated in the PSB field.
	LOGMES	With variant X.C (CICS multi-screen) and 3.X (Cobol Microfocus) : generates a logical message which will manage both screen display and message sending to the communications monitor selected with the MONIT option (if it is not a default value in the variant).
		IMS: The LOGMES option is used only for the Web.
	LPAGE LPA	LOGICAL PAGING management in implicit mode (with Function Key PA1) or in explicit mode (with a Logical Paging Operator on the Screen Call of Elements (-CE) Screen). (See chapter 'Definition and Description of a Dialogue or Screen', subchapter 'Screen Call of Elements').
		The LPAGE option is available with a monitor, a sub-monitor, or with MONITOFF.
		With the Monitoff option, pagination can only be done on the same screens.
		With a (sub-)monitor pagination is possible on several screens by entering the screen codes on the Dialogue's Work Areas (-W) screen. For more information, please refer to the IMS-DB/DC 'On-Line Systems' manual, chapter 'The Generated Program', subchapter 'Additional Screens'.
		NOTE: If using sub-monitors, all the screens that make up a pagination group must be in the same sub-monitor.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The LPAGE and SWAP options are not compatible. If they are both coded, only the LPAGE option is recognized.
	MONIT	UNISYS Series A: generation with monitor.
		IBM CICS: generation with monitor.
	MONITOFF MOFF	IMS: One transaction code per dialogue screen and no monitor generation.
		MONITOFF and NOTRAC options are incompatible. If both are entered, MONITOFF overrides NOTRAC.
	NOATTR	IMS: Generates protected fields in the MOD without their attributes (ATTR=YES). This option is valid with all the screens of a dialogue.
	NOLOCK	AS400: reading of the Segments without update and with addition of the WITH NO LOCK clause in the reading function when the file is open in OPEN I-O.
		This option does not work if several segments of a same file are accessed, some with update, some without, since READ WITH NO LOCK removes the preceding lock.
	NOOPEN	MICROFOCUS and DEC: optimization of the opens of HE files (Help file backup) and LE or EM (error messages) and of the CONNECT to ORACLE.
	NOOPENB	HP3000, TurboImage: this option stops the opening of the TurboImage database. The OPEN is not generated in the program.
	NOPSB	Segment generation limitation:
		In each screen, only those segments used plus their hierarchical parents are generated. In the monitor and sub-monitors, the PSB field is no longer generated nor transmitted to the screens.
	NOSAV	With DEC/VAX, GCOS7, MICROFOCUS, ICL, IBM 36 and 38, and BOS/TP: Avoids the generation of the description of, and the physical access to, the HELP file (HE).
	NOSDERR	UNISYS 2200: recognition of the ERMSG Data Element in the message, which therefore is not sent by the SEND ERROR clause.
	NOTRAC	IMS: No generation of transaction code in the MID of the first screen of the dialogue.
	OCF	The OCF variable is set to value '1' at the first program run (Current default value).
	OFF	The dialogue's programs and maps are generated with option 'MDT-OFF' (default value for Gcos7 FORMS).
	ON	Default value (except for Gcos7 FORMS). The dialogue's programs and maps are generated with option MDT-ON.
	PACSQL	Validation of the data entered in the -GG of the Segment for the SQL orders (refer to the 'SQL Databases' manual, chapter 'SQL Accesses', subchapter 'Customized SQL Accesses').
	PSBINIT	CICS-DL1 screen with monitor option: this option allows to perform the connection to the PSB and its closing (SCHEDULE PSB and TERMINATE) on the sub-monitor and not on each screen.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	REFER REF	Generates the list of all Data Elements used in a dialogue. This list is found in the WORKING-STORAGE SECTION and has a documentary value. It includes:
		- Data elements from segments called on the On-Line Screen Call of Segments (-CS) screen,
		- Data Elements from Segments called in the WORKING-STORAGE SECTION on F-type Work Areas (-W) lines,
		- Data Elements directly called in the WORKING-STORAGE SECTION on I-, E-, or S-type Work Areas (-W) lines,
		- Data Elements directly called on the On-Line Screen Call of Elements (-CE) screen.
	REPET RPT	The validation, transfer and positioning of attributes is taken into account for variable fields repeated horizontally and/or vertically.
		When a repeated field is used in a Segment access, you must indicate the number of repeated uses at the Segment call level of the related screen.
	SQLALL SQA	SQL: Allows all the table columns to be enumerated in the generated commands for segment access.
		(SELECT ALL COLCORUBn, ...
		INTO :FFNN-CORUBn)
	SQLCA SQC	SQL: Generation of the SQL command 'INCLUDE SQLCA' in WORKING-STORAGE if there is a least one DB2 or Non-Stop SQL Segment called in the screen: ('-CE' screen, ORGANIZATION = 'S', 'Q' or 'N')
	SQLIND SQI	Generation of presence indicators for the Data Element prior to update or display.
	SQLREF SQR	Indicator generation with the new V-FFnn-DELCO structure, allowing automatic management of the cross-references.
	SSMON	The SSMON option allows the coding of one PSB for each sub-monitor instead of one PSB for all the screens in the dialogue. (Refer to the 'Generation Elements' screen, -GG).
	SWAP	IMS: Option to be used with a (sub-)monitor.
		The SWAP option allows the code of the screen to branch to via a Function Key to be indicated in the MFS map. In each map, a 10-character FUNC field is generated and includes the PFKEY value and the external name of the next screen to branch to.
		For End-Of-Conversation, the external name of the screen must be left blank.
		When using deferred branching, or setting OPER with the PFKEY, the external name is that of the generated map.
		The FUNC field is processed in the (sub-)monitor.
		The SWAP option improves performance as it avoids loading the program which processes the branching.
		NOTE: If the MONITOFF or LPAGE option is selected, the SWAP option is not taken into account.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	TMF	TANDEM: With this option it is possible to take into account the transactional mode through the generation of BEGIN-TRANSACTION and END-TRANSACTION orders in the requesters.
		BOS/TP: With this option it is possible to take into account global transactions through the generation of 'tpbegin', 'tpcommit', and 'tpabort' orders in the clients.
	TS	CICS: Allows for the management of the Communication Area outside of the LINKAGE SECTION. This area is described in WORKING and saved in TEMPORARY STORAGE by default. In addition, in case of MDTOFF and when the HELP function is called, the screen is saved in a unique area.
	TZVAR	BULL: With this option, it is possible to fill in the reception and display areas in case of error.
	ZNUM	The transfer of numeric fields in the message (F8145) is conditioned by the fact that no error was encountered (SCR-ER = 0).

---

## Overriding Generated Constants (-GG)

The 'Generation Elements' screen (-GG) is used in relation to a given Screen or Dialogue (for example: for error messages). You may also use this screen to override some of the values of the generated constants.

### CODING

To code an overriding line, use the 'G'-type line in the 'C2' option (O: C2) (the same as when using P.I.A.'s).

Each line is composed of two fields:

- A left-justified two-character code in the DESCRIPTION field, according to the hardware in use. Refer to the CODE LIST below.
- The new value justified on the tabulation position in the center of the DESCRIPTION field.

### CODE LIST

- 01: IMS

Modification of the value of the parameter 'TYPE' (Terminal type) in the macro-instruction DEV of the MFS format description.

(Default value: TYPE=(3270,2)).

MULTI-TERMINAL DIALOGUES:

The same screen description can be generated for different terminal types:

```
A LIN : T COMMENT
      100 : G 01 n           3270-A1
      200 : G 01 n           3270-A2
with n = 0-9
```

- 02: IMS

Modification of the value of the 'FILL' parameter in the macro-instruction MFLD of the MOD.

(Default value: FILL=PT).

- **03: ALL HARDWARE**  
Modification of the external name of the HELP function display program.  
(Default value: PACHELP).  
NOTE: This external name can be entered on the HELP Screen Definition screen.  
Thus, this option should be disregarded.
- **04: ALL HARDWARE**  
Modification of the external name of the program which manages the accesses to the Pactables function.  
(Default value: PACTABLE).
- **05: GCOS7, GCOS7 FORMS**  
Modification of the name of the screen backup file when HELP function is called:  
(Default value: HE)  
DEC/VAX, ICL, IBM 36, IBM 38/AS 400, MICROFOCUS, DM6 TP and BOS/TP:  
Modification of the name of the screen backup file for HELP function calls. The new value must be coded as:  
XX EXTERN  
where: XX = name of the file and EXTERN = external name of this file; two blank characters must be inserted between these two names.  
(Default value: HE SAVESCR).
- **06: CICS**  
Modification of the first four characters of the name of the TS used by the HELP function.  
(Default value: PAC7).
- **07: GCOS7**  
Value used to replace blanks in output messages.  
(Default value: 19).
- **08: GCOS7 FORMS**  
Modification of the value 'NORMAL' of the intensity attribute.  
(Default value: SPACE).
- **09: GCOS7 FORMS**  
Modification of the value 'BRIGHT' of the intensity attribute.  
(Default value: BD).
- **10: GCOS7 FORMS**  
Modification of the value 'DARK' of the intensity attribute.  
(Default value: CN).
- **11: CICS DL/1: HLPI**  
Database Lock shift (Y/N) when access with update.  
(Default value: LOCKED).
- **12: DEC/VAX, IBM CICS, DPS7 and DPS8 with the MULTI-SCREEN type transaction variant, ICL, MICROFOCUS, BOS/TP:**  
Modification of the name of the subprogram called for message reception and formatting.  
(Default value: ZAR980;  
MICROFOCUS: ZAR980.EXE in DOS, ZAR980.DLL in OS/2, UNIX).

For IBM CICS, ZAR980 corresponds to user program PRUSER; to replace the standard PRCGI, see parameter number 22.

For Pacbase Web Connection, use ZAR990 program which is the same as ZAR980 but with a web specificity.

- 13: GCOS8  
Modification of the number of characters sent in each 'DISPLAY' for the message display.  
(Default value: 2,000).
- 14: DEC/VAX  
Modification of the number of characters sent in each message 'DISPLAY' (when there are too many characters, another 'DISPLAY' is executed).  
(Default value: 1,000)
- 15: IBM 36, IBM 38/AS 400:  
DISPLAY FILE external name.  
(Default value: SCRFILE).
- 16: CICS  
Option to write the TEMPORARY STORAGE in AUXILIARY STORAGE ('AUXILIARY') instead of the MAIN STORAGE.  
(Default value: MAIN)
- 17: ALL HARDWARE  
Character string (11 characters maximum) for the HELP Screen title.  
(Default value: SPACE).
- 18: ICL  
Name of file used by the monitor in order to save the communication area.  
For Pacbase Web Connection, all hardware except IMS.  
Name of the contextual backup file for a WEB connection.  
The new value must be coded as in parameter 05. Example: WE SAVEWEB, where WEB corresponds to the name of the file in the program and SAVEWEB corresponds to the external name of the file.
- 19: IMS  
At the dialogue or sub-monitor level, for screens using DB2 segments:  
For each DBRM: external name of the screen (8 characters).  
The lines which include this parameter are grouped as comments at the beginning of the WORKING-STORAGE SECTION of the generated (sub-)monitor.
- 20: IMS  
External name of the module which replaces the standard module (external name: 'CALL') in monitors and sub-monitors.  
NOTE: With COBOL II, if this parameter is not entered, a dynamic COBOL CALL is generated.
- 21: IBM CICS, IBM 38/AS 400, UNISYS 2200, TANDEM, GCOS7 FORMS  
This parameter takes the values 'ARRAY' or 'TABLE' if a field is repeated horizontally AND vertically on the screen. The logical message will be initialized from the physical message, so that they will match.
- 22: IBM CICS  
For the multi-terminal type variant, name of the sub-program to replace the standard 'PRCGI' sub-program (external name D4R980) that formats a '3270' type message.



- 26: UNISYS 2200  
Enables you to replace the default intensity attribute NORMAL with LOW, since some screens do not make any difference between the NORMAL and HIGH intensity values. As the attributes for label fields cannot be assigned at the elementary fields level, the attributes defined on the screen definition-screen for the label fields must be added to the FORM NAME format. The ATTRE option must be specified on the Dialogue complement.
  - 27: IBM 38/AS 400  
Indicator number for the first line of a repetitive category in case of error. For the following lines, the indicator is incremented by 1.  
Example: specified value is 34; each data element in the first line of the repetitive category will be assigned indicator 34, those on the second line, indicator 35.
  - 28: IBM 38/AS 400  
Recognition of lowercase characters in the DDS of the screen formats of clause CHGINPDFT (LC). Value LC entered after the second tabulation in the DESCRIPTION field must be typed in uppercase characters.
  - 29: TANDEM Requester  
Generation of the ESCAPE ON clause on the screen's ACCEPT. (Default value = SPACE). Information specified for this parameter (up to 40 characters) will be added after the ESCAPE ON clause.
  - 30: TANDEM Requester  
Assignment of a function key for end of input in the case where the PFKey is not used as operation code.
  - 31: CICS  
MAP BMS: Parameters modification of the CTRL option contained in the definition of the MAP SET (DFHMSD).  
By default, the generation is:  
CTRL=(FREEKB,FRSET)  
The modified parameters are parenthesized instead of FREEKB,FRSET.
  - 32: UNISYS-A  
Insertion of working descriptions between the I-O-MESSAGE area and the other areas of the linkage section.  
To realize this insertion, you must indicate the location of the dialogue linkage description. This description is generated by default from line 00001 onwards. The two first figures of this line number can now be parameterized.
  - 33: BULL  
Parameterization of the key loading of the backup file for the help function.  
By default, the key is loaded from the terminal code defined in the TRANSACTION-STORAGE. To parameterize the key loading, the parameter 33 is associated to the Data Division area in which you initialize the key (for example USER-ID defined in the TDS-STORAGE).  
You must therefore code, in the -G screen of the dialogue:  

33	USER-ID
----	---------
- The name of this area is limited to 14 characters. The default value of this parameter is 'XTERM' and the key loading is performed from the K-Sscre-XTERM area.
- 34: WEB CONNECTION, MICROFOCUS, TANDEM AND UNISYS, WITH A nnnn FORMAT.

Parameter used to indicate the size of the backup file in the common area.

- 35: ALL HARDWARE FOR A WEB CONNECTION:  
Length of the logical message (5 characters). The default value of this parameter is 14000. It does not include the length of the constant part of the message.
- 36: COMMUNICATION TYPE  
Communication for CICS:
  - ECI: the default communication monitor generated.
  - CPIC: CPIC LU 6.2 communication.
  - SOCKET: TCP-IP
  - MQSERIESCommunication for IMS:
  - CPIC: default value
  - MQSERIESCommunication for Gcos7 and Gcos8:
  - XCP2: default valueCommunication for MICROFOCUS, Tandem and Unisys:
  - SOCKET: TCP-IP, default value
- 37: LENGTH OF THE COMMUNICATION BUFFER WITH A nnnnn FORMAT.  
CICS and IMS MQSERIES: 14200 as default value for MQSERIES.  
CICS SOCKET: 02048  
GCOS7 XCP2: 4092  
GCOS8 XCP2: 4092  
TANDEM SOCKET: 14200 as default value
- 38: TIMEOUT  
Maximum time of wait for a client communication (format: nnnn) in seconds.  
CICS:
  - SOCKET:1800 (default value)CICS, IMS:
  - MQSERIES: unlimited as default value.
- 39: UNISYS  
Name of backup file in the Dialogue context.  
Format: 40 characters length

#### IMS: GENERATION OF A PSB PER SUB-MONITOR

In order to implement this option, the SSMON value must be entered on the Dialogue Complement (-O) screen in the OPTIONS field.

In order to generate a PSB per sub-monitor (as opposed to a general PSB for the whole dialogue), on the 'Generation Elements' screen (-GG) of each sub-monitor, you must specify on a 'G'-type line, in C2 (OPERATION):

- PSB= left-justified in the DESCRIPTION field,
- PSB code, at the tabulation position in the center of the DESCRIPTION field.

This line must be the first -GG line.

**NOTE:** Each screen must belong to only ONE sub-monitor.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1	ACTION CODE (REQUIRED)
	C	Creation of the line
	M	Modification of the line
	D	Deletion of the line
	A	Deletion of the line
	T	Transfer of the line
	B	Beginning of multiple deletion
	G	Multiple transfer
	?	Request for HELP documentation
	E or -	Inhibit implicit update
	X	Implicit update without upper/lowercase processing (on certain lines only)
		On the GP-C4 screen (JCL command lines), upper/lowercase processing.
		On the GP-C1 screen, upper/lowercase processing on continuation lines only.
2	3	Line number
		PURE NUMERIC FIELD
		It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
3	1	LINE TYPE - GENERATION ELEMENTS
	G	You use a 'G' ligne to:
		. generate, from a Segment, a customized SQL access with a delimiter.
		. indicate the physical characteristics of a Block on the Block Definition.
		. add generation data on the Block Description.
		. override generated constants in a Screen or TUI Client.
	V	Generate, from a Segment, a customized SQL access without a delimiter.
	I	Call of a P.I.A.
		You can only call a P.I.A. with a 'G' type.
		The lines from the called P.I.A. have a 'Line generation option' set to 'blank' or 'G' in the P.I.A. Description.
		The call line is then replaced by the called P.I.A.
		Socrate Databases
	V	Line taken into account at generation.
	P	Line taken into account at generation.
	Z	Line taken into account at generation.
4	60	DESCRIPTION / FIRST PART
		If no P.I.A. is called, you can use the whole field.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			The contents of this field vary according to the type of line and to the current entity.
			See the corresponding manuals.
5	40		DESCRIPTION / SECOND PART
			This field is specific to a P.I.A. call.
			With value 'C2' in the OPERATION CODE field, the cursor automatically tabs to the first position of this field.
			This field is initialized with underscores (default value) or with the value specified in the INITIAL VALUE field for a Standard PIA description line (Type = 'blank').
			If symbolic parameters have been defined on the P.I.A. Description (-D), they may be entered in this field. They will be replaced by their corresponding value, and will remain displayed on the right of the screen.

---

## Screen Call of Segments (-CS)

This screen is used to indicate how to access Segments in the validation, update and display processing associated with the screen.

The number of Segment description lines called in this screen is limited to 100 per category.

Segments having the same Data Structure codes must have the same organizations, the same types of description, and, with a V-type organization, the same external names.

For a D-type organization, segments having the same Data structure codes are generated in REDEFINES.

### GENERAL CHARACTERISTICS

For each Segment, the following attributes are specified:

- The category in which the Segment is being processed,
- The type of logical access to the Segment in Reception and/or Display, and the filling mode of the key,
- Use in generated procedures (for validation, update, display),
- The sequence in which the Segments are to be accessed,
- Physical organization.

It is possible to enter a particular type of logical access for a given Segment for each type of use (validation, update, display).

The segment codes that are being used to describe the screen (Screen Call of Elements) are displayed at the top of the screen.

The organization of a segment, together with the variant of the TP monitor indicated on the Screen Definition screen, results in the generation of the segment description in the DATA DIVISION.

For a Segment whose access is determined by specific procedures (ORGANIZATION = 'U'), the description will be generated in the WORKING-STORAGE SECTION of the program.

WITH DL/1:

Since the System automatically deduces the key filling mode in DL/1 (qualified SSA), it is not possible to use the same key Data Element codes for different segments of the same hierarchical sequence within a given PCB (parent segments).

>>>> If you use the VisualAge Pacbase WorkStation, the graphical interface of the corresponding windows is described in the 'WorkStation User Interface' guide.

### GENERATION OF ACCESS PROCEDURES

The logical accesses to a Segment are generated independently of the variant, depending on how the Segment is used (in reception or in display).

A segment call automatically generates a number of lines in programs that can be grouped in the following way:

- Segment description including:
  - Segment I/O areas,
  - DATA DIVISION.
- Logical Access Procedures:
  - Read of files in Reception (F25),
  - Write of files (F35),
  - Access display files (F60).
- Physical access: (F80).

These three groups of lines are generated or not according to the values of the following three fields:

- GENERATION OF LOGICAL ACCESSES OF SEGMENTS,
- ORGANIZATION OF THE SEGMENT,
- USE IN RECEPTION OR DISPLAY.

Values 'U', 'W', 'T' and 'A' in the ORGANIZATION OF THE SEGMENT field suppress the generation of function F80.

Value 'U' in the GENERATION OF LOGICAL ACCESS TO SEG. field suppresses the generation of functions F25, F35 and F60 depending upon the USE IN RECEPTION or DISPLAY field values. The value 'P' in the GENERATION OF LOGICAL ACCESS TO SEG. field suppresses the generation of the description part of the Segment.

With value 'D' in the ORGANIZATION OF THE SEGMENT field (DL1), if several Segments have the same parent segment in the PSB, all accesses to these segments use the same SSA from the parent Segment.

### PREREQUISITE DESCRIPTIONS

The screen and the called Segments must have been defined.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6	SCREEN CODE (REQUIRED)
		All six positions of the screen code are required. The first two characters must correspond to the code of the dialogue to which the screen belongs.
		The dialogue must have been previously defined.
2	1	ACTION CODE (REQUIRED)
	C	Creation of the line
	M	Modification of the line
	D	Deletion of the line
	A	Deletion of the line
	T	Transfer of the line
	B	Beginning of multiple deletion
	G	Multiple transfer
	?	Request for HELP documentation
	E or -	Inhibit implicit update
	X	Implicit update without upper/lowercase processing (on certain lines only)
		On the GP-C4 screen (JCL command lines), upper/lowercase processing.
		On the GP-C1 screen, upper/lowercase processing on continuation lines only.
3	4	SEGMENT CODE IN THE PROGRAM (REQUIRED)
	WE00	Segment code used by the generated Program associated to the Screen. The Segment code used can be identical to the one present in the Database. Value used for the -CS screen of the IMS Web communications monitor.
4	1	CATEGORY INDICATOR
		Used to indicate the screen category in which this Segment is being processed.
		The category, and therefore the corresponding Segments are processed in the following order:
		.The screen-top category.
		.The repetitive category as many times as there are repetitions indicated.
		.The screen-bottom category.
	blank	Segment belonging to the screen-top.
	R	Segment belonging to the repetitive section.
	Z	Segment belonging to the screen-bottom.
5	2	NUMER.
		PURE NUMERIC FIELD
	00	Only one '00' line number per Segment is permitted. (Default)

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The values entered at this level are the physical characteristics of the Segment (external name, organization), the name of the Segment access key and the name of the Segment in the database library.
	01 - 99	Required for any other references to this Segment.
		All other characteristics (Use, Access Key Source, etc..) can be entered on these continuation lines.
6	1	GENERATION LIMITATION
	blank	Logical access generated automatically, according to the use of the Segment.
	U	User-defined Segment access.
		NOTE: Segment I/O areas are automatically defined (IMS SSA's).
	P	No description of Segment.
7	1	USE IN RECEPTION
		Used to specify the way in which the Segment is used in validation and update processing.
	N	Not used.
		No access statement will be generated for this Segment in the validation and update processing.
	L	Read.
		A simple read command is generated for the Segment.
	E	Existence validation to execute.
		A read command with the setting of an error code is generated (if it is not found).
	M	Updated Segment (modification only).
	X	Updated Segment (creation and modification only).
	T	Updated Segment (creation, modification and deletion).
		In the last three cases, the existence validations and the update statements corresponding to the chosen options are generated.
		If the Segment is related to a PRECEDING SEGMENT, regardless of the indicated use, only the update commands for modification and creation are generated.
		NOTE: For Segments accesses conditioned by a PRECEDING SEGMENT access, update commands for deletions will not be generated.
8	1	USE IN DISPLAY
	N	Not used.
		No access will be generated for this Segment in the display processing (default option).
	A	Used in display.
		The access statement will be generated for the display according to the type of logical access processing.
		If the PRECEDING SEGMENT code is blank and the Segment is used, the key is saved in the conversation area.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
9	4		PRECEDING SEGMENT
			This is the code of the Segment which is to be accessed prior to the segment entered in the SEGMENT CODE IN THE PROGRAM field.
			If the access of the PRECEDING SEGMENT is unsuccessful, the Segment is bypassed.
			If several segments have a common PRECEDING SEGMENT, they will be processed according to the sequence of their codes.
			NOTES: It is not recommended to link Server calls and calls of Segments which correspond to DataBases or files.
10	14		ACCESS KEY SOURCE
			This field indicates the source to be used to fill in the key fields of the Segment.
			EXAMPLE:
			.MA60-DATEL for a field on the screen.
			.'02 ' for a literal.
			.WW10-DATEL for a work area.
		*SSA	Entered on the '00' line, the '*SSA' value inhibits the generation of the qualified SSA corresponding to the Segment in the PCB.
			The next field, i.e. ACCESS KEY (DATA ELEMENT CODE), must be entered with the code of the Data Element used as the display key. The K-.... field is generated with the access key indicated in the ACCESS KEY (DATA ELEMENT CODE) field. The key will be filled in automatically in functions F25, F40, and F60.
			The SSA of the processed Segment is filled in from the '-CS' lines whose key code has not been found in the hierarchical path.
			NOTE: If the Segment is used in display in the repetitive category, the K-....-.....(2) key is not filled in since the Segment's access key is not defined.
			This specification is not available with the HLPI option (the key is directly coded in the WHERE parameter).
11	6		ACCESS KEY (DATA ELEMENT CODE)
			This is the DATA ELEMENT CODE used as the access key to a Segment.
			For 'V' and 'W' organizations in OLSD function, it must be entered on the Segment's '00' line.
			The Data Element on this line is considered as a group Data Element, except in SQL organizations where it is considered as an elementary Data Element.
			If a group Data Element is used, and different values are given to the individual members of that group, the codes of those elementary Data Elements are named here.
			For a Table ('G' ORGANIZATION), selection of a sub- system is executed by indicating the value 'NUSSY' in this field. The system generates the associated key (G-ddss-NUSSY).



NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			In DL/1, the Data Element can belong to all SSA's that permit access to the Segment in the PCB indicated in the external name (code of the PCB).
			SQL Database Segments
			A key Element must be entered on each line where the value of the USE IN DISPLAY field is not 'N'.
			In generated accesses, the Segments are selected when the keys are equal for update or a direct read; for a sequential read in the repetitive category, the Segments are selected when their keys are equal or greater.
			A Segment can not contain more than 10 key Data Elements.
			Key Data Elements are not updated in a SQL UPDATE order.
12	1		CONTROL BREAK INDICATOR FOR DISPLAY
		blank	No control break.
		C	Display break. This provides an indication of the logical end-of-file for control break processing in the repetitive section . Control break processing will take place under either of these conditions:
			1. The value in the field of the group key Data Element identified with this 'C' is different from the previous value in this field.
			2. The value of a superior key Data Element does not match its previous value.
			SQL ORGANIZATIONS:
		E	Display control break; causes the generation of an equal condition in the WHERE clause of the SQL 'DE- CLARE CURSOR' statement (and not >=, which is gene- rated otherwise). Allows optimization of the use of CURSORS in the repetitive category.
13	1		ORGANIZATION OF THE SEGMENT (REQUIRED IN CREAT)
			Required for the '00' LINE NUMBER of the Segment.
		V	Segment Organization: Indexed.
		D	DL/1 Segment.
			A data description is generated for the two organizations mentioned above. The Segment access for the update and display processing will be generated according to the type of logical access processing and use.
		G	'TABLE'-type Segment. In this case, the generated access corresponds to the call of the general module of the Pactables function.
		A	'ADABAS'-type Segment. In this case, the Segment description is generated, but no access is executed.
		T	'TOTAL'-type Segment. In this case, the Segment description is generated, but no access is executed.
		2	DB2 or VAX SQL Segment generation and description. Only local accesses are not generated. The structure of variable indicators corresponding to the columns of a DB2 or VAX SQL table is always generated.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Segment of a Relational Database accessed via SQL: The Segment description and physical accesses are generated. For segment access in the repetitive category, the cursor declarative is also generated.
		C	INTEREL RDBC or RFM
		O	Oracle V5
		P	Oracle V6 or V7
		Q	SQL/DS, ALLBASE/SQL, DB2/2 or DB2/6000
		S	DB2 or VAX SQL
		R	RDMS
		4	SQL/400
		7	TurboImage
		N	NONSTOP SQL
		M	DATAKOM/DB
		I	INFORMIX, INGRES, SYBASE
			NOTE: Relational organization must be grouped together in the -CS screen.
		U	Physical access to the Segment managed by specific procedures.
		W	Segment access is managed by specific procedures and the display keys are automatically filled in (functions F40 and F60).
		Y	Database organization. Value used with IBM SYSTEM 38. UNISYS 2200: SFS indexed organization.
14	1		DESCRIPTION TYPE
			This field defines the type of Segment being used.
		blank	Conventional and multi-record files of fixed length: this type of Segment will be appended to the '00' (common key) Segment.
		1	Database Segments -- those that do not have their keys isolated in a separate Segment.
			(Default value for a DL/1 Segment, ORGANIZATION 'D' or for a table, ORGANIZATION 'G'.)
		E	TABLE-type Segments: The input format of the Segment's Data Elements is taken into account.
		2	* Conventional and multi-record files of variable length: this type Segment will be appended to the '00' (common key) Segment.
			* For Type P organization (ORACLE V6) and Type I organization (SYBASE): automatic generation of CONNECT AT Database, DECLARE Database and access SQL AT Database.
			Note that the generated description may vary according to the generation variant.
15	8		EXTERNAL NAME OF THE FILE
			(Default value: first two characters of the Segment code in the program.)
			IBM CICS-VSAM: DDNAME (1 to 8 positions).
			DL/1: PCB name.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		SQL: Code of the SQL Block in which the Table or View is called. It is possible to select a Table or a View by specifying a 'T' or 'V' after the code of the selected Block. A space must be left between them.
		At generation time, the external name of the Table or View is searched for in the Block description. If it is not found, the external name is taken from the Segment Definition (value in the CODE/VALUE OF RECORD TYPE ELEMENT field).
		GCOS7 - UFAS: External name of the file (two characters).
		The System generates the following clause:
		COPY SELECT XX-FILE (XX = ext. name).
		GCOS8 - TP8: External name on four characters (generated in the SELECT clause).
		Blocking factor on last four characters (generated in the FD clause).
		OTHER VARIANTS: Logical name, generated in the SELECT clause.
		'G' ORGANIZATION (Tables defined to Pactables):
		In CICS: 4 characters for the DDNAMEs of the files managed by the Pactables function (these are the characters of the transaction code used to access Pactables files). Enter the 4 character DDNAME of your table file.
		In IMS: Enter the first 1 to 6 characters of the external names you assigned to the description and value PCB's on the Database Block Definition screen.
		NOTE: Do not include the 'D' or 'V' in this field.
		Example: if the description and value PCB's have been defined with their external names as 'TBLPCBD' and 'TBLPCBV' respectively, this field would have the value 'TBLPCB'.
16	4	SEGMENT CODE
		Field to be entered when the SEGMENT CODE IN THE SCREEN is different from the Segment code stated at the Specifications Dictionary level.
17	1	SUB-SCHEMA NUMBER
		'G' or 'H' organizations:
		For Tables defined in Pactables, this specifies the number of the Table sub-description (or sub-schema) to which the input-output area description corresponds. If the Segment called corresponds to a View and no sub-schema has been specified, the value will be that specified on the Block '-DR'.
		BLANK
		All the Data Elements of the Segment.
		0 to 9
		Sub-description (or sub-schema) number (1 to 10, where value 0 corresponds to sub-schema 10).
		V-type organizations: Secondary access keys to indexed files:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		2	The secondary key is specified with the value '2' in the SUB-SCHEMA NUMBER field. The primary key must be indicated on line '00' of the Segment without use in display or reception, even if it is not used, in order to generate the RECORD KEY clause.
			If the secondary key is a group area, the number of the sub-schema must only be indicated for this group area.
		3	The value 3 indicates that the secondary key is DUPLICATE.
			NOTE: This specification is not implemented in the CICS variant because the declaration of the secondary keys is performed at the VSAM definition.
18	2		LEVEL OF GENERATED SUB-FUNCTIONS
			PURE NUMERIC FIELD
		06 to 98	The On-line Systems Development function automatically generates the following in the program associated with the screen:
			.A sub-function of Function 20 for each Data Element to be validated,
			.A sub-function of Functions 25, 35 and 60 for each Segment which is accessed in the screen.
			See the 'Structured Code' manual for the definitions of function, sub-function and level notions. Refer to subchapter 'Standard Structure of the Procedure Division' in the Appendix of the 'On-Line Systems Development' manual for the list of functions of the generated program.
			The hierarchical level is 10 by default. It can be modified for a Data Element or a Segment.

## DL/1 Secondary Index Management

The generation of Segment accesses via secondary indexes is based on the description of Database Blocks in the Data Dictionary.

This generation option has the following characteristics:

- The SHARING SECONDARY INDEX DATA BASEs are not processed and consequently, neither is the INDICES parameter.
- The PROCSEQ parameter, followed by the external name of the INDEX DATA BASE, must be indicated in the 'Generation Elements' screen (-GG) associated with the PCB call line in a PSB.
- If the external name coded on the Segment call line is a physical DBD, the Segment for this DBD must have the LCHILD and XDFLD macros specified on the 'Generation Elements' screen.
- If the external name coded on the Segment call line is a PCB, the external name coded on the PCB Definition screen must correspond to the external name of the corresponding DBD, so that the LCHILD and XDFLD macros can be found for the Segment containing a secondary index.
- The NAME parameter must be entered on the same line as LCHILD or XDFLD.

- The SRCH parameter and XDFLD may be entered on different lines. However, the fields which make up the SRCH parameter must be on the same line as the SRCH.
- The SEGMENT parameter and XDFLD may be entered on different lines.
- There is no display break on a Data Element of the source Segment if it is different from the target Segment, since the key does not belong to the target Segment.
- The filling in of the keys on the '-CS' screen is done with either the key field codes specified in the SRCH parameter of the XDFLD macro, or the group field generated in the SSA to which the keys belong (CORUB).  
The code of this group field must be indicated in the ACCESS KEY (DATA ELEMENT CODE) field on the '-CS' screen.  
CORUBn (n = numeric indicator) is generated for Element display keys, and K-FFNN-... is generated for the Segments used in display; its length will be the sum of the lengths of each of the key fields.  
If a PACBASE group field is used as a key field, it is not split into elementary Elements.
- Only the SSA's, whose key indicator corresponds to the indicator entered on the PCB description (-DH), will be generated (and not an SSA for each access key entered on the Segment Call of Elements (-CE) screen).

The code 'x' of the generated SSA 'S-FFxNN' will be the one that must be indicated when calling the Segment into the PCB (or DBD).

The Source Segment code must be called in a PCB of the Screen or Dialogue's PSB so that the description of its elements can be retrieved

**NOTES:** When the secondary access is done on a Segment which is used in display in the repetitive category, the access key K-Rddss-CORUB (2) is not filled in in F60, since it is not always part of the Target Segment.

There is a limitation of 200 secondary indexes per screen.



---

## Chapter 6. Use of Structured Code

---

### Introduction

The Structured Code is a high-level language which is used to develop programs and systems.

Used in conjunction with the Specifications Dictionary, Structured Code offers the following advantages:

- Improved conciseness in the writing of COBOL programs, due to the simplification of COBOL commands and syntax,
- Hierarchical structuring of procedures, which does not exist in COBOL,
- Structured programming using the following types of structures: BLOCK (BL), IF THEN (IT) ELSE (EL), DO WHILE (DW), DO UNTIL (DU), CASE OF (CO),
- Improved portability, due to independence from any given hardware and the ability to generate in a COBOL code adapted to each environment.

Structured Code also provides facilities to define Work Areas (-W), and common processing routines can be called into an On-line program, using the '-CP' screen.

See subchapter 'Purpose of the Manual' in the 'Introduction' chapter for references to the other manuals where entities used by the On-Line Systems Development function are mentioned.

This chapter covers the specifics of using Structured Code to write specific procedures (-P) in conjunction with the OLSD function.

---

### Specific Procedures

Automatic functions generated for a dialogue give standard solutions to standard problems, but do not always respond to all the processing needs of a given user.

You can determine, at the development stage of an application, the way in which you wish to solve your problem. It is recommended to use the standard solutions provided by automatic functions as often as possible.

However, it is not recommended to try to fit a very complex procedure into the automatic functions if it is going to detract from the clear structure of the program. It is very important to completely think through the specifics of the required procedures when a program is being developed.

If you have special needs, you may complete or partially replace certain standard procedures using the Structured Code function.

The Structured Code is used to enter detailed specific procedures. Its use ensures the structuring of procedures, as well as the readability and convertibility of programs.

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

>>>> If you use the VisualAge Pacbase WorkStation, the graphical interface of the corresponding windows is described in the 'WorkStation User Interface' guide.

## SPECIFIC OPERATORS FOR ON-LINE SYSTEMS DEVELOPMENT

**'Yaa'**: Generates a COBOL paragraph label for Function 80. Followed by a segment code (ddss) in the OPERAND field, will generate a 'F80-ddss-aa' COBOL paragraph label.

**'Xaa'**: generates a PERFORM of paragraph 'F80-ddss-aa'.

### OPERATORS FOR ERROR POSITIONING:

**'ERR'**: Error on Data Element.

**'ERU'**: Dialogue-related error.

For more details concerning these operators, see chapter 'Error Messages - Help Function', subchapter 'User-Defined Error Messages'.

### OPERATORS FOR END-OF-PROCESSING:

**'GFA'**: End of display processing. (Branch to 'END-OF-DISPLAY' paragraph)

**'GFR'**: End of reception processing. (Branch to 'END-OF-RECEPTION' paragraph)

**'GDB'**: Return to the beginning of current iteration.

**'GDI'**: Go to the beginning of next iteration. (Branch to next occurrence of current category or to next category)

**'GFT'**: Go to the end of the iteration. (Branch to the end of category processing)

**'GF'**: Go to end of automatic sub-function that has been entered by using a type '\*A' (or '\*P' if the level code of the inserted function is greater than that of the generated one).

**WARNING:** With the 'GFR', 'GFA' and 'GF' OPERATORS, the generated instruction is always followed by a full stop (.). This means that no 'EL' (ELSE) type of conditioning should be used on a following line, as the generated COBOL would then be erroneous.

### OPERATORS FOR SCREEN CONCATENATION:

**'OTP'**: Call of the screen whose external name is indicated as the OPERAND).

**'OSC'**: Call of the screen (code) indicated as the OPERAND.

**'OSD'**: Call of the screen (code) indicated as the OPERAND (deferred to the end-of-reception processing).

For more information on the operators for date processing, see the 'Structured Code' manual, chapter 'Modifying the procedure division', subchapter 'Procedural code screen (-P)'.

## INSERTING AND REPLACING AUTOMATIC SUB-FUNCTIONS

On the first line of a processing block, the CONDITION TYPE OR S.F. STRUCTURE is indicated.



EXAMPLES: 'BL' = Block, 'IT' = If Then, 'EL' = Else, 'DW' = Do While, etc.

These block types conform to the rules of structured programming.

The block type is used in the OLSF function to insert specific procedures into the automatic functions: F20, F25, F30, F60, F65 and F80.

These procedures are written in one or more sub-functions with the same function code as the function into which they are to be inserted (F20sf, F25sf, F30sf, F60sf, F65sf, and F80sf).

The sub-function code ('sf') must not take on the value of any of the automatically generated sub-functions, which always have a numeric value. It is therefore recommended to systematically use an alphabetic sub-function ('sf') code for the procedures to be inserted.

Function F20 contains a sub-function for each edited Data Element. The sub-function into which the specific procedures are to be inserted is identified by the code of the corresponding Data Element.

Functions F20, F35 and F60 contain a sub-function for each accessed Segment. The sub-function into which the specific procedures are to be inserted is identified by the code of the corresponding Segment, followed by the code of the category, if the same Segment code is to be used in several categories.

Functions F30 and F65 contain a sub-function for each processed category. The sub-function into which the specific procedures are to be inserted is identified by the code of the corresponding category.

Function F80, when generated, includes a sub-function for each type of access of a given Segment. It is possible to override one particular generated access, or override all generated accesses of a given Segment.

Indicate in the CONDITION FOR EXECUTION field, left-justified:

- For F20, the Data Element Code.
- For F25, F35 or F60, the Segment Code and possibly the category code (Screen-top = ' ', Repetitive = 'R', Screen-bottom = 'Z').

EX: 10\*A CD10 R

and in the BLOCK TYPE field:

- '\*A' for the insertion of the sub-function before the automatic sub-function.
- '\*P' for the insertion of the sub-function after the automatic sub-function. The condition of the automatic sub-function applies to the inserted sub-function if the LEVEL NUMBER of the inserted sub-function is greater than that of the automatic sub-function.
- '\*R' to replace an automatic sub-function. The condition of the automatic sub-function does not apply to the replaced sub-function.
- For F30 or F65, the category code. (Screen-top = ' ', Repetitive = 'R', Screen Bottom = 'Z')

EX: 10\*P R

and in the BLOCK TYPE field:

- '\*P' in order to insert a sub-function after the automatically generated sub-function.

The condition of the automatic sub-function does apply to the replaced sub-function.

- For F80, the Segment code, followed by the type of access to be replaced.  
Ex: '10\*R CD10 RN'  
and in the BLOCK TYPE field:
- '\*R' to replace one or all of the generated physical accesses for this Segment.

For these three types, the sub-function code does not have any influence on the placement of the sub-function in the generated program, but is used only for its classification in the libraries.

### REPLACEMENT OF PROCEDURAL CODE IN F80

To replace the procedural code in function 80, indicate the Segment code in the \*R type Block and in the condition field.

If the Segment code is followed by the type of access, only the sub-function corresponding to the access type will be replaced; if the access type is not indicated, all the access functions to this Segment will be replaced.

Examples:	10*R FF00	-->	Replacement of all the access to Segment FF00.
	10*R FF00 R	-->	Replacement of the read of Segment FF00.

For complete information on the access types and on their customization, see the 'SQL Databases' manual, chapter 'SQL Accesses', subchapter 'Customized SQL Accesses'.

---

## Chapter 7. Pacbase Web Connection Implementation

---

### Pacbase Web Connection

#### GENERAL PRINCIPLES

Pacbase Web Connection allows you to revamp applications developed with the OLSD module in the HTML page format. The description of the dialogue must include some pieces of information relating to the communication.

In the host part, you must generate a logical message which will allow screen display management or message sending to a Web communication monitor.

For the first communication, the monitor calls the first screen and creates, in the form of file, a backup field for the common area of the screen.

For the following transactions, the common area is retrieved, then the screen program is called.

For each return from the program screen, the monitor sends the message to the Web and saves the common area. Each screen program is a sub-program of the monitor.

#### DIALOGUE

##### LOGMES OPTION:

OLSD programs are adapted to the Web communication through the exchange of a logical message. The description of this logical message is obtained when you specify the LOGMES option in the -O screen.

##### GENERATION VARIANT:

- Use a multi-screen variant in the Dialogue Definition Screen if this variant is available for your material.

##### MONIT OPTION:

- Specify the MONIT option in -O if it is not set by default.

>>>> For more details on these options, refer to chapter 'Description of a transaction', subchapter 'Dialogue complement (-O)'.

#### GENERATION

Use the GEO command, option C4 to generate the file which contains the Dialogue description.

Then, you must extract from the refential the file containing the description of the screens that must be revamped and transfer them on the computer where Pacbase Web Generator is installed.

>>>> For more information, refer to the 'Pacbase Web Connection' manual in the Developer's Documentation.

## COMMUNICATIONS MONITOR

The communications monitor used for a connection to the Web is specified via an instance of the MW-type screen entity.

>>>> For more details, refer to chapter 'Definition and Description of a Dialogue or Screen', subchapter 'Dialogue or Screen Definition (O....)'.

In the -GG screen of the monitor, you must specify:

- the name of the file used to save the communication area (parameter 18),
- the length of the backup field in the common area (parameter 34 for MICROFOCUS, Tandem and Unisys).
- the length of the logical message (parameter 35),
- the communication type (parameter 36),
- the length of the buffer (parameter 37),
- the timeout (parameter 38).
- 2 - the name of the file used to save the Dialogue context (parameter 39 for Unisys).

>>>> These parameters are described in chapter 'Description of a Transaction', subchapter 'Overriding Generated Constants'.

FOR IMS:

Call in the -CS screen of the communication monitor, the WE00 segment so as to save the Dialog context.

The segment is made of the following fields:

- a KEY field which consists on 3 fields:
  - IDENT (26 characters),
  - STRUCT (1 character),
  - NUM (4 characters).
- a CAREA field of n characters length, where n must be higher or equal to the size of the Dialogue context.

WE00 is the name that must be used in the -CS screen of the web communication monitor.

>>>> For more information on how to fill the fields in -CD Screen, refer to chapter 'Description of a Transaction', subchapter 'Screen Call of Segments (-CS)'.

---

## Structure of the Monitor

```
First transaction
    Initialization
    Context backup
    First screen
Following transactions
    Context retrieval
Screen program call
Return
    Error
    End of transaction
    Screen linking
    Message sending
```

Context backup



---

## Chapter 8. Error Messages - Help Function

---

### Error Messages - Introduction

The System manages error messages that will be used to inform you of input errors detected by application programs.

Error messages can be created as needed, or generated upon request, to update the sequential error message file. This file will be used to create application error message files. They can be indexed files or databases, depending on the hardware in use.

The generation is performed by the GPRT procedure, using the GEO print-generation command. It generates the error messages for the screens specified in the GEO command inside the PAC7GL file. Error messages of other screens found in the PAC7LG file are copied in the PAC7GL file and not modified.

#### AUTOMATIC ERROR MESSAGES

There is an automatic error message which corresponds to each validation instruction (entered on the Screen Call of Elements) and to each file access.

Error messages are divided into two parts:

- A message specifying the error type and thus related to the validation or file access type. These standard error messages are stored in one of the System files (however, they can be modified on-site by the Database Administrator).
- The Segment or Data Element name (except for Segments used in display).

#### DATA ELEMENTS

The System generates four standard error messages for validations on Data Elements:

```
'INVALID ABSENCE FOR THE FIELD'  
'INVALID PRESENCE FOR THE FIELD'  
'NON-NUMERICAL/NON-ALPHABETICAL CLASS FIELD'  
'INVALID VALUE FOR THE FIELD'
```

#### SEGMENTS

The System generates the following standard file access error messages:

- Segments used for input:  
'INVALID CREATION RECORD',  
'INVALID MODIFY/DELETE RECORD',

Both messages include the Segment's name.

- Segments used for display:  
'END OF LIST'

These automatically generated error messages may be customized to your needs.

Overriding the standard error messages is done on the 'Error Messages-Help' screen (-GE). For more details, please refer to subchapter 'Overriding Automatic Error Messages'.

### EXPLICIT ERROR MESSAGES

Only validations (entered on the Screen Call of Elements) and file accesses trigger the automatic generation of error messages. For other error types detected through other validations, error messages must be explicitly entered. This is done on the 'Error messages-Help' screen (-GE screen) associated with either the considered Screen or Dialogue, depending on the type of error.

---

## **Error Messages - Coding**

The -GE screen, 'Error Messages-Help' screen, is used to replace error messages generated by the System, to declare explicit error messages, or to enter comments for error messages.

The DESCRIPTION field on this screen is coded according to the TYPE OF LINE value:

F = Data Element call,

S = Segment call,

U = Error message.

C = Comment(s).

---

## **Overriding Automatic Error Messages**

Overriding an automatically generated error message is done on the 'Error Messages-Help' screen (-GE).

Two line types must be described:

- . TYPE OF LINE value = 'F' for Data Element call, or  
= 'S' for Segment call.
- . TYPE OF LINE value = 'U' for error message.

Several 'U'-type lines may follow 'F'- or 'S'-type lines.

And possibly:

- . TYPE OF LINE value = 'C' for comments,  
attached to the error message which directly precedes it.

### DATA ELEMENTS

- . TYPE OF LINE value = 'F' (Data Element call)

Positions in the DESCRIPTION field:

1 to 6 : Data Element code.

- . TYPE OF LINE value = 'U' (error message)

Positions in the DESCRIPTION field:

1 to 3 : blank

4 : number of error type to be replaced for  
the Data Element (2, 3, 4 or 5)



```

        5 : blank
        6 : error message.
. TYPE OF LINE value = 'C' (comments on the error message)
Positions :
    1 to 5 : blank
    6 : the comment(s).
EXAMPLE: F NUCLI
        U   2 CLIENT NUMBER IS REQUIRED
        U   5 CLIENT NUMBER MUST BE > 10001

```

## SEGMENTS

The OLSD function generates error codes and corresponding messages for file access errors.

You may override these messages on the -GE screen assigned to the screen. Two types of lines must be described: type 'S' and type 'U'. Several type 'U' lines can follow a single type 'S' line.

```

. TYPE OF LINE value = 'S' (segment call)
Positions :
    1 to 4 : Segment code
    5 : blank
    6 : category code (required if the same seg-
        ment code is used in several categories):
        Screen-top : ' ', Repetitive : 'R',
        Screen-bottom : 'Z'
    7 : Segment position in the category (required if
        the same Segment code is used several times in
        a category).
. TYPE OF LINE value = 'U' (error message)
Positions :
    1 : Segment use category
        'F' for reception
        'G' for display
    2 to 3 : blank
    4 : error type code
        '8' record found, (invalid creation)
        '9' record not found, (invalid modification,
            end of file, etc.)
    5 : blank
    6 : error message.
. TYPE OF LINE value = 'C' (comment)
Positions :
    1 to 5 : blank
    6 : the comment(s).
EX : S CD10
        U F 8 PRODUCT REFERENCE NUMBER ALREADY ASSIGNED
        U G 9 END OF SEARCH FOR THIS STATE

```

## NOTES

It is possible to replace an automatic error message on a single line, on the -GE screen. On a 'U'-type line, the rank of a Data Element on the Screen Call of Elements must be entered in positions 1 - 3; the rank of a Segment on the Screen Call of Segments must be entered in positions 2 - 3.

The rest of the DESCRIPTION field is entered as indicated above. Please note that this method implies recoding of error codes when a Data Element is added to or deleted from the Call of Elements screen, or when a Segment is added to or deleted from the the Call of Segments Screen

---

## User-Defined Error Messages

There are two types of explicit (or user-defined) error messages:

- Errors on Data Elements,
- Dialogue-related errors.

### 1. ERRORS ON DATA ELEMENTS

A Data Element error is described on the 'Error Messages-Help' screen' (-GE).

Two types of lines must be described: an 'F'-type and a 'U'-type. Several 'U'-type lines may follow an 'F'-type line. It is also possible to specify 'C'-type lines.

```
. TYPE OF LINE value = 'F' (Data Element call)
Positions :
  1 to 6 : Data Element code.
. TYPE OF LINE value = 'U' (error message)
Positions :
  1 to 3 : blank
  4 : error code (alphanumeric character
    except for '0' and '1')
  5 : blank
  6 : error message.
```

```
EX: F NUCLI
    U   3 THIS CLIENT NUMBER IS NOT AUTHORIZED
    U   A THE TYPE DOES NOT CORRESPOND TO THIS CLIENT
```

### PROCEDURAL CODE (-P) LINES CODING REMINDERS

The ERR operator causes a field level error message to be generated.

OPERATOR: ERR

OPERAND field:

Positions :

- 1 : Code of the error (alphanumeric character except for '0' and '1' which are reserved for coding documentary messages)
- 2 : blank
- 3 : Code (six positions) of the variable Data Element to which the error code is associated. The cursor is positioned and the Data Element takes on the attributes previously defined for erroneous Data Elements. With a repetitive Data Element, its code must be entered and has to be followed by the sequence number of the Data Element occurrence.

EXAMPLE: DELCO (2)

## 2. DIALOGUE-RELATED ERRORS

A dialogue-related error is described on the -GE screen using 'U'-type lines.

. TYPE OF LINE value = 'U' (error message)

Positions :

1 to 4 : code of the error (mentioned on a '-P' line).  
5 : blank  
6 : error message

EXAMPLE:

U AV01 DATE AND NATURE OF NOTICE MUST CORRESPOND

## PROCEDURAL CODE (-P) LINES CODING REMINDERS

The ERU operator causes a Dialogue-related error message to be generated.

OPERATOR: ERU

OPERAND field:

Positions :

1 to 4 : error code (user-defined)  
5 : blank  
6 : code of the variable Data Element (optional).

The error message corresponding to the error code is displayed on the error message line (ERMSG).

If a Data Element code has been entered, the cursor is positioned on this Data Element, and error attributes apply.

**NOTE:** The ERU operator cannot be used on a repetitive Data Element.

Specific errors are coded in Procedural Code lines (CH: O.....P). The corresponding error messages are entered according to rules specified in this subchapter.

Since the character '\*\*' is preceded and followed by space in lines -P, it is not recommended to use it in an Error Message file key.

---

## Help Function - Introduction

The Help function allows to branch to a Help documentation screen at any time.

Help documentation is present at two levels:

- Screen-level Help (how to use the screen),
- Input field Help (for example, possible values).

You branch to a Help screen by positioning the cursor on the appropriate field and pressing a PFkey, or by positioning a designated character in the CHOICE field for screen-level Help, or in the desired input field for input field Help.

On the Help screen, the CHOICE allows you to:

- Return to the beginning of the Help documentation,
- Continue scrolling through all Help documentation,
- To stop the Help display and return to the initial work screen.

The Definition screen of the Help screen must be created, but it is not described, since its format is fixed and cannot be modified.

The code of this screen will be XXHELP, where XX is the Dialogue Code; the external name must also be indicated ('PACHELP' by default). It must then be generated using the appropriate TYPE OF COBOL/TP MONITOR, MAP TO GENERATE variants, according to the hardware in use.

Information concerning generation and links with other screens in a Dialogue can be found in the 'On-Line Systems Development' manual specific to the hardware in use.

---

## Help Messages - Introduction

The System allows for the generation of documentary Help messages that will be used to document Screens and Data Elements and to explain the rules for information coding and error detection.

These messages can be created as needed and generated on request to update the sequential error message file.

This file will be used to create the applications' error messages files. These are indexed files or databases depending on the hardware in use.

The error messages file of a given application will be accessible in the generated transactions and for documentary purposes via the standard program performing the Help function.

The screen-level Help documentation is taken from:

- the 'Error Messages-Help' screen (O.....GE).

The input field Help documentation is taken from:

- the Data Element Description (E.....D) in the Dictionary
- Documentation of the specific screen Data Element (text or comments associated with the Data Element on the Error messages-Help (-GE) screen; 'C' or 'T' lines following a 'F' line.
- Standard error messages (type '2' to '5')
- User error messages (alphabetic type 'A' to 'Z'),
- User error messages (numeric type '6' to '9').
- Description lines or text calls associated with Data Element-related errors on the Error messages-Help (-GE) screen.
  - User errors:  
'C' or 'T' lines following a 'U' line.
  - User or standard errors:  
'C' or 'T' lines with the error code.

**NOTE:** In order to have a better presentation of the HELP screen, the numeric type user error messages should be used rather than the alphabetic type, as the numeric type are grouped together with the standard error messages.

---

## Help Messages - Coding

Error messages are described on the 'Error Messages-Help' screen (-GE).

Help Messages lines are coded as follows:

- TYPE OF LINE value 'C' = comments
- TYPE OF LINE value 'T' = text call

In the case of a called text, the coding of the 'T' type line is as follows:

Positions:

6 - 11: code of called text

12 - 13: code of the selected paragraph, or '\*\*' to select all paragraphs.

### NOTES

#### 1. TEXT PROCESSING

The text calls for the documentation are aligned according to a standard print layout.

All the Text calls are justified (except if the non-justification '\$S' parameter was used). The '\$n' print windows indicated in the text are respected. If they are not specified, the default print window ('\$0') is used.

All the text processing possibilities are functional (underlining, negative indentation, etc.).

Line skips are specified in the text, but limited to a three-line skip maximum (line skips greater than three become three-line skips); page skips become three-line skips.

By default, the skips are one line, except for print windows \$0 and \$1, where they are two lines.

Thus, the same text can be a part of the 'on-line' documentation and an application user manual.

For complete information on text processing in relation to the Help function, see the 'Personalized Documentation Manager (PDM)' manual.

#### 2. GENERAL DIALOGUE ERRORS

You can assign comment lines to explicit manual errors on the -GE screen associated with the dialogue. These lines will not be accessible, from the standard HELP function program, in the generated transactions.

### SCREEN-RELATED DOCUMENTATION

The documentary lines associated assigned to a screen are found on the the 'Error messages-Help' screen (-GE) associated with the screen.

Positions:

1 to 5: blank

6: the comments

on a 'T'-type line,  
6-11: code of the called text line  
12-13: paragraph code ('\*\*' for all paragraphs).

Help screens related to a screen include:

- Screen 'C'- or 'T'-type lines,
- Segment 'U'-, 'C'-, or 'T'-type lines.

### DOCUMENTATION OF A DATA ELEMENT

Data Element 'C' or 'T'-type lines must follow an 'F'-type line referring to the same Data Element.

Positions:

- 1 to 3: blank
- 4: '0' (zero)
- 5: blank
- 6: comments, or on a 'T'-type line the code of the called text, followed by the paragraph code, or '\*\*' for all paragraphs.

The Data Element documentation that you can access via the on-line Help in the generated transactions, is made up of 'C'- or 'T'-type General Documentation lines assigned to the Data Element, Data Element description lines (blank-type), error messages generated for the given Data Element, and, if appropriate, their related documentation lines.

### DOCUMENTATION LINES OF ERROR MESSAGES

#### 1. EXPLICIT ERROR MESSAGES (EXCEPT ERRORS ON DATA ELEMENT)

Positions:

- 1 to 4: ERROR CODE:  
BLANK if following a 'U', 'C' or 'T' -type line relating to the error.

#### 2. ERROR WITH AN AUTOMATIC MESSAGE OR ERROR ON DATA ELEMENT

Positions:

- 1 to 3: ERROR CODE:  
BLANK if following an 'F'-, 'U'-, 'C'- or 'T'-type line.
- 4: Position of the documentary message:  
'2 4 5 6 7 8 ...' comments on the line after the error message of the indicated type (2 4 5 6 7 8 ...)
- 5: BLANK
- 6: comments, or in a 'T'-type line the code of the called text, followed by the paragraph code, or '\*\*' for all paragraphs.

## Coding Screen

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1		ACTION CODE (REQUIRED)
		C	Creation of the line
		M	Modification of the line
		D	Deletion of the line
		A	Deletion of the line
		T	Transfer of the line
		B	Beginning of multiple deletion
		G	Multiple transfer
		?	Request for HELP documentation
		E or -	Inhibit implicit update
		X	Implicit update without upper/lowercase processing (on certain lines only)
			On the GP-C4 screen (JCL command lines), upper/lowercase processing.
			On the GP-C1 screen, upper/lowercase processing on continuation lines only.
2	3		Line number
			PURE NUMERIC FIELD
			It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
3	1		LINE TYPE - MESSAGES GENERATION
		T	Call of a Text section.
			PACBENCH C/S AND ON-LINE SYSTEMS DEVELOPMENT FUNCTIONS
		S	Segment call.
		F	Data Element call.
		U	Error message.
		C	Comment. See the 'Pacbench C/S Business Logic and TUI Clients' manual or the 'OLSD' manual, chapter 'Error Messages'.
			BATCH SYSTEMS DEVELOPMENT FUNCTION
		S	Replacement of an automatic error message
		D	Documentary messages. See the 'Batch Applications' manual, chapter 'Error Messages'.
4	60		DESCRIPTION
			BATCH SYSTEMS DEVELOPMENT FUNCTION
			DOCUMENTARY MESSAGE: 'D'-type line.
			Position 1:
			'0': Message before Element Description.
			'1': Message after Element Description.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'2-5': Message after an error message of type 2 to 5.
		Position 2:
		blank.
		Position 3:
		'blank': Message entered on the line.
		'T': Call of a Text instance.
		Position 5-...Documentary message, or Text & Paragraph codes (**: ALL paragraphs).
		OVERRIDING A STANDARD ERROR MESSAGE: 'S'-type line.
		Position 1:
		'2-5': Error type.
		Position 2: blank.
		Position 3: Error gravity (E, C or W).
		Position 4: Blank.
		Position 5-...: Error message.
		PACBENCH C/S AND ON-LINE SYSTEMS DEVELOPMENT FUNCTIONS
		SCREEN-RELATED DOCUMENTATION: 'C'- or 'T'-type line.
		Position 1-5: Blank.
		Position 6-...: Message on 'C'-type line or Text & Paragraph codes on 'T'-type line.
		OVERRIDING A DATA ELEMENT-RELATED STANDARD ERROR MESSAGE OR CREATING A DATA ELEMENT-RELATED USER-DEFINED ERROR MESSAGE:
		NOTE: Two lines need be coded.
		1. 'F'-type line:
		Position 1-6: Data Element instance code.
		2. 'U'-type line:
		Position 1-3: Blank.
		Position 4:
		'2-5': Standard error type.
		'N' : Code of manual error (save '0' and '1')
		Position 5: Blank.
		Position 6-...: Error message.
		OVERRIDING STANDARD SEGMENT-RELATED ERROR MESSAGES: NOTE: Two lines need be coded.
		1. 'S'-type line:
		Position 1-4: Segment instance code.
		Position 5: Blank.
		Position 6: Category:
		'A' Heading,
		'R' Repetitive,



NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		'Z' Bottom.
		Position 7: Segment's rank in the category (if used several times in the category).
		2. 'U'-type line:
		Position 1:
		'F': Segment used in reception.
		'G': Segment used in display.
		Position 2-3: Blank.
		Position 4:
		'8': Invalid segment creation.
		'9': Segment not found.
		Position 5: Blank.
		Position 6-...: Error message.
		USER-DEFINED DIALOGUE-RELATED ERROR MESSAGE: 'U'-type line.
		Position 1-4: Error code.
		Position 5: Blank.
		Position 6-...: Error message.
		DOCUMENTATION OF A DATA ELEMENT:
		Position 1-3: Blank.
		Position 4: '0' (zero).
		Position 5: Blank.
		Position 6-...: Documentation lines, or Text and Paragraph codes after a 'T'-type line.
		DOCUMENTATION OF A DIALOGUE-RELATED ERROR MESSAGE:
		Position 1-4: Error code. 'blank' If following a 'U'-, 'C'-, or 'T'-type line.
		Position 5: Blank.
		Position 6-...: Documentation lines, or Text and Paragraph codes after a 'T'-type line.
		DOCUMENTATION OF A STANDARD ERROR MESSAGE AND OF A USER-DEFINED DATA ELEMENT-RELATED ERROR MESSAGE:
		COL. VALUE DESCRIPTION
		Position 1-3: Error code.
		'blank' If following an 'F'-, 'U'-, 'C'-, or 'T'-type line.
		Position 4: Type of error whose message is to be documented.
		Position 5: Blank.
		Position 6-...: Documentation lines, or Text and Paragraph codes after a 'T'-type line.
5	40	DESCRIPTION / SECOND PART

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		You can tabulate directly to this field. However, it has not any special use in this screen.

---

## Message Generation and/or Printing

GEO : Generation and printing of error messages.

OPTION C1 - Error messages defined for the dialogue and for each screen.

OPTION C2 - Error messages generated through option 1 plus documentary help messages.

OPTION C3 - Error messages for the dialogue only.

OPTION C4 - Generation of the file containing the description of the screens to be 'revamped' with Pacbase Web Connection.

>>>> Refer to the 'User Interface Guide' for information on the GPRT procedure and the 'Web Connection Operator's guide' for information on the procedure output, the transfer procedure and the application revamping.

The error messages language may be chosen by entering a code in column 28 of the \* line of the GPRT procedure: E for English, F for French.

If you do not enter a code here, the language code of the database (specified in the REST procedure) will be used as default language code for the error messages.

LEO: List the error messages defined for the dialogue and for each screen. This list only includes messages that have already been generated.

This command is accessible in option 1 only.

**NOTE:** If a screen suffix is entered on the continuation line of a GEO or LEO command, error messages are generated/printed for this screen only.

UPC - UPPERCASE SHIFT: If the printer in use does not process lowercase characters, the 'UPC' command allows for the automatic transformation of lowercase characters into uppercase characters when reports are printed.

- On the 'GP' screen:

When the UPC command is entered, the following label is displayed:

SHIFT TO UPPERCASE MANUAL: \_ DOC: \_ ERROR MESS:

The value '1' must be entered in the ERROR MESS field in order to request an uppercase shift.

- In batch mode:

Enter '1' in column 21.

---

## Examples and Limitations

### EXAMPLE OF ERROR MESSAGE OUTPUT

```

+-----+-----+-----+-----+
!ERR G! LIST OF ERROR MESSAGES          !
+-----+-----+-----+-----+
!      !                               !
!      ! NUMBER OF DELIVERIES          !
!      ! -----                          !

```

```

!      ! Lines of text or documentation associated with the !
!      ! Data Element.                                     !
!      !                                                    !
!      ! Data Element description lines.                   !
!      ! 0          .Before creation of the first delivery. !
!      ! 1 to 9    .At each creation of a delivery, the value !
!      !           is incremented by 1.                    !
!      !                                                    !
!0082E!INVALID ABSENCE OF DATA ELEMENT NUMBER OF DELIVERIES!
!      !                                                    !
!0084E!NON NUMERIC CLASS DATA ELEMENT  NUMBER OF DELIVERIES!
!      !                                                    !
!      ! Lines of text or documentation associated with Data!
!      ! Element error type '4'.                           !
!      !                                                    !
!0085E!ERRONEOUS VALUE OF DATA ELEMENT NUMBER OF DELIVERIES!
!      !                                                    !
!0086E! SINGLE DELIVERY FOR THIS CLIENT                    !
!      !                                                    !
!      ! Lines of text or documentation associated with !
!      ! Data Element error type '6'.                      !
+-----+-----+-----+-----+-----+-----+-----+-----+

```

### LIMITATION

Up to 999 'S', 'F', 'U', 'C' or 'T'-type General Documentation lines can be associated with a given screen.

## Description of the Error Message File

The System generates an error message file. The records generated for this file are described on the following pages.

Example of error message file records:

```

-----
! GCCHJIE0100054000ENON-NUMERIC CLASS DATA ELEMENT ACTION !
!                                                                !
! LU1ID0000116  002 009                                     !
-----

```

Decoding the first example:

```

LIBRARY CODE : CGC
ENTITY TYPE  : H
ENTITY CODE  : JIE010
ERROR NUMBER : 0054
LINE NUMBER  : 000
ERROR GRAVITY: E
ERROR MESSAGE: NON-NUMERIC CLASS DATA.....

```

This file has the following physical characteristics:

```

Organization: indexed
Access key   : position 1
               17 characters
Records      : 90 characters

```

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	3	LIBRARY CODE
2	1	ENTITY TYPE
		Used to specify the type of entity.
	A	For data structures or Segments (BSD error messages).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	H	For screens (OLSD error messages).
	I	Record reserved for internal use by the OLSD function. It is used by the Help function to indicate the position of a field on a screen, using a line / column formula.
3	6	ENTITY CODE
4	4	ERROR NUMBER
		There are three ways to build the error number:
		1. AUTOMATIC ERROR MESSAGE, or EXPLICIT MANUAL ERROR MESSAGE 'ON A DATA ELEMENT',
		The first three positions identify the rank of the Data Element causing the error, by the order in which it was defined on the Call of Elements (-CE) screen. (Example: the fifth Data Element defined)
		NOTE: When this screen is modified, the rank may also be modified, changing the element to which this error message refers.
		Position 4 is defined as follows:
	2	. invalid absence
	4	. erroneous class
	5	. erroneous value
	6 7 ...	. additional errors
		2. DOCUMENTARY MESSAGE ASSOCIATED TO AN AUTOMATIC ERROR MESSAGE OR MANUAL ERROR MESSAGES 'ON THE DATA ELEMENT'
		The first three positions identify the rank of the Data Element causing the error, by the order in which it was defined on the Call of Elements (-CE) screen. (Example: the fifth Data Element defined)
		NOTE: When this screen is modified, the rank may also be modified, changing the Element to which this error message refers.
		Position 4 is defined as follows:
	0	Documentation will be generated before the data element description lines.
	Other	Documentation will be generated after the error message of the indicated type.
		3. ASSOCIATED EXPLICIT AND DOCUMENTARY MESSAGES
		Number of the automatic explicit error message, or number of the manual explicit error message.
		NOTE: For an ENTITY TYPE 'T' record, this number is managed by the system and contains the LINE NUMBER of the erroneous field on the screen.
5	3	LINE NUMBER
		This number is managed by the system.
	000	Error messages
	001-999	Documentary messages

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		NOTE: For an ENTITY TYPE 'I' record, this number is managed by the system and contains the LINE NUMBER of the erroneous field on the screen.
6	1	SEVERITY OF ERROR
		A standard error gravity of 'E' is associated with each automatic or explicit error message.
		A standard error gravity of 'W' is associated with each warning entered for an error message.
7	30	ERROR MESSAGE FIRST PART
		This part of the message indicates the error type for automatic error messages in the following manner:
		2 INVALID ABSENCE FOR THE FIELD
		4 NON-NUMERICAL CLASS FIELD
		5 INVALID VALUE FOR THE FIELD
		It can be modified.
		For an explicit error message, this is the first part of the message indicated in the DESCRIPTION field on the -GE screen (first 30 positions).
		NOTE: For an ENTITY TYPE 'I' record, this field is managed by the system and contains the position of the corresponding Data Element in the description of the screen.
8	36	ERROR MESSAGE SECOND PART
		For an automatic error message, this is the name of the Data Element as defined in the Data Dictionary.
		For an explicit error message, this is the second part of the message in the DESCRIPTION field on the -GE screen (position 31 to the end).



---

## Chapter 9. Access Commands

---

### On-Line Access Commands

#### SCREENS : CHOICES

##### LIST OF SCREENS

CHOICE	SCREEN	UPD
-----	-----	---
LC0aabbbb	List of screens by code (starting with screen 'aabbbb').	NO
LP0ddddddd	List of screens by external name (starting with ext. name 'ddddddd').	NO
LT0ttaabbbb	List of screens by type ('tt' is the type and 'aabbbb' the screen code).	NO
LS0mmmmmm	List of screens by external map (starting with map 'mmmmmm').	NO
LN0ttOeeeeee	List of screens by type ( 'tt' is the type and 'eeeeee' is the first screen code in the list)	NO
LOTttttttt	List by external transaction name (starting with transac. name 'ttttttt').	NO

##### DESCRIPTION OF SCREEN 'aabbbb'

CHOICE	SCREEN	UPD
-----	-----	---
0aabbbb	Definition of On-Line Screen 'aabbbb'.	YES
0aabbbbGCccc	Comments for Screen 'aabbbb' (starting with line 'ccc').	YES
0aabbbbGEccc	Error Messages - Help on screen 'aabbbb' (starting with line 'ccc').	YES
0aabbbbG0ccc	Generation Options of Screen 'aabbbb' (starting with line 'ccc').	YES
0aabbbbGGccc	Generation Elements of Screen 'aabbbb' (starting with line 'ccc').	YES
0aabbbbATcccccc	Text assigned to Screen 'aabbbb' (starting with text 'cccccc').	NO
0aabbbbX	X-references of Screen 'aabbbb'.	NO
0aabbbbX0	X-references of Screen 'aabbbb' to screens	NO
0aabbbbXQrrrrrr	List of occurrences linked to Screen 'aabbbb' through User Relationship 'rrrrrr'.	NO
0aabbbbCRrrrrrr	Occurrences linked to Screen 'aabbbb' through User Relationship 'rrrrrr'.	YES
0aabbbbXVvvvvvv	X-references of Screen 'aabbbb' to Documents (starting with Document 'vvvvvv').	NO
0aabbbbCEccc	Data Elements of Screen 'aabbbb' (starting from line 'ccc').	YES

OaabbbbCSddss	Call of Segments of Screen 'aabbbb' (starting with category indicator 'c' and Segment ddss). NOTE: 'c' is equivalent to '&' for the Screen-top category.	YES
OaabbbbCPccccc	Call of Parameterized Macro-Structures in Screen 'aabbbb' (starting with Macro-Structure 'ccccc').	YES
OaabbbbBccddee	Beginning Insertions Modifications for on-line Screen 'aabbbb' (starting with section 'cc', paragraph 'dd', line 'eee').	YES
OaabbbbWccddd	Description of Work Areas (-W) of Screen 'aabbbb' (starting with Work Area 'cc', line 'ddd').	YES
OaabbbbPfusfnn	Description of Procedural Code (-P) of Screen 'aabbbb' (starting with function 'fu', sub-function 'sf', line number 'nn').	YES
aabbbbPGfusfnn	View of Procedures Generated of Screen 'aabbbb' (starting with function 'fu', sub-function 'sf', line number 'nn'), with display of generated procedure titles.	YES
OaabbbbLnnCxxx	Layout of Screen 'aabbbb' (starting with line 'nn', column 'xxx').	YES
OaabbbbMnnCxxx	Mapping of Screen 'aabbbb' (starting with line 'nn', column 'xxx').	YES
Oaabbbb0	Complement of Dialogue definition file 'aa' (also obtained without specifying 'bbbb').	YES
OaabbbbT0fusf<nn	View of Titles Only of Screen 'aabbbb' (starting with function 'fu', sub-function 'sf' up to level 'nn').	NO
OaabbbbTCfusf<nn	View of Titles and Conditions of Screen 'aabbbb' (starting with function 'fu', sub-function 'sf' up to level 'nn').	YES
Oaabbbb<nn	View of Titles and Conditions of automatic and specific procedures of Screen 'aabbbb' (up to level 'nn').	YES
OaabbbbADR11Cnn	Data Elements of Dialogue 'aabbbb' with their positions, from line '11', column 'nnn'.	NO
OaabbbbSIMnn	Dialogue simulation from Screen 'aabbbb'. 'nn' specifies the end of simulation (function key on most hardware). If 'nn' is not specified, the simulation ends on a 'Clear Screen'.	NO

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

All notations between parentheses are optional.



---

## Screen Cross-References

A number of entities are used in the different screens that describe a Screen entity. These uses, i.e. cross-references, are listed below.

### SCREEN CROSS-REFERENCES TO SCREENS (CH: O.....XO)

- Screen Title or Contents Call ('-CE'),
- Automatic Branchings ('-CE'),
- Usage of OSC or OSD operator ('-P').

### DATA ELEMENT CROSS-REFERENCES TO SCREENS (CH: E.....XO)

- Data Element Call in Layout ('-CE'),
- Operand or Condition ('-P'),  
(including in the Operand of the ERU and ERR operators or  
in the Condition for sub-functions of type \*A, \*P, or \*R)
- Data Element Call in Working Area ('-W'),
- Error Message Definition ('-GE').

### SEGMENT CROSS-REFERENCES TO SCREENS (CH: S.....XO)

- Segment Call ('-CS'),
- Segment Call in Working ('-W').

### DATABASE BLOCK CROSS-REFERENCES TO SCREENS (CH: B.....XO)

- External name of Block (ORGANIZATIOn = 'D') ('-CS'),
- PSB External Name ('-O' and '-GG', containing 'PSB = ...')

### PROGRAM CROSS-REFERENCES TO SCREENS (CH: P.....XO)

- Macro-Structure Calls ('-CP').

---

## Generation and/or Printing

From the Generation and Print Commands (GP) screen, you can request either screen descriptions (DCO command), or screen description and generation (GCO command). Depending on which OPERATION code is entered, the requested output is:

### SCREEN DESCRIPTION: DCO

- OPTION C1 - description of Screen layout
  - description of Segments
  - description of Structured code
- OPTION C2 - description of Screen layout
  - description of assigned text
- OPTION C3 - description of Screen layout
  - description of assigned text
  - description of Segments
  - description of structured code

### SCREEN DESCRIPTION AND GENERATION: GCO

OPTION C1 - description of Screen layout - description of Segments - description of structured code - screen generation

OPTION C2 - description of Screen layout or C3 - description of assigned text - description of Segments - description of structured code - Screen generation



---

## Chapter 10. Appendix

---

### Standard Structure of the Procedure Division

---

#### STANDARD STRUCTURE OF THE PROCEDURE DIVISION

F0110 Initializations

-----  
F05 RECEPTION (ICF = '1')  
F0510 Reception of the screen  
F0512 Documentation call procedure  
F0520 Validation of Operation Code (OPER)  
F1010 Category processing <-----  
F15 Validation of the Transaction Code (CATM)!  
F20 Data element validation !  
F25 Segment access for validation !  
F30 Data element transfer !  
F35 Segment access for update !  
F3999-ITER-FN. GO TO F10. -----  
F3999-ITER-FT. EXIT.  
F40 END-OF-RECEPTION PROCESSING  
F4010 Set-up keys for new display  
F4020 Set-up keys for screen paging  
F4030 End of transaction  
F4040 Transfer to another screen  
END-OF-RECEPTION. (F45-FN)

-----  
F50 DISPLAY PREPARATION (OCF = '1')  
F5010 Initialization  
F5510 Category processing <-----  
F60 Segment access for display !  
F65 Data element transfer !  
F6999-ITER-FN. GO TO F55. -----  
F6999-ITER-FT. EXIT.  
F7010 Error processing  
F7020 Positioning of attributes  
END-OF-DISPLAY. (F78-FN)

-----  
F8Z05 Memorization of the screen  
F8Z10 Send of screen  
F8Z20 End of program  
----- Called functions -----  
F80 Physical access to files  
F8095 Save of screen (documentation call)  
F8098 Error Message File Access  
F81ER Abnormal end of procedure  
F81UT Memorization of errors for display  
F8110 Numeric class validation  
F8115 Initialization of the variable fields  
F8120 Date format validation  
F8125 Transfer to display  
F8130 Help function procedure  
F8135 Transfer to reception  
F8140 Cursor position calculation

NOTE:

Function F81 can contain other sub-functions, notably those for message management. Their generation depends on the on-line monitor in use and the generation variant.

Refer to the 'On-Line Systems Development' manual specific to your target environment.

---

## Chart of Constants and Variables

```

+-----+
!           CHART OF ON-LINE CONSTANTS AND VARIABLES           !
+-----+
!           !           !           !           !           !
! CURPOS ! CURSOR POSITIONING IN RECEPTION SCREEN WHERE           !
!           ! CPOSL = LINE NUMBER & CPOSC = COLUMN NUMBER           !
!           ! (except for GCOS7 FORMS).                       !
!           !           !           !           !           !
! CPOSN  ! 'ABSOLUTE' CURSOR POSITIONING WHERE CPOSL = 1           !
!           ! AND CPOSC = 1                                           !
!           ! (except for GCOS7 FORMS).                       !
!           !           !           !           !           !
! INA    ! Input field.                                           !
!           ! NUMBER OF DATA ELEMENTS IN SCREEN-TOP CATEGORY !
!           !           !           !           !           !
! INR    ! Input field.                                           !
!           ! INA + NUMBER OF DATA ELEMENT IN REPETITIVE           !
!           ! CATEGORY.                                           !
!           !           !           !           !           !
! INZ    ! Input field.                                           !
!           ! INR + NUMBER OF DATA ELEMENTS IN SCREEN-BOTTOM !
!           ! CATEGORY.                                           !
!           !           !           !           !           !
! IRR    ! NUMBER OF REPETITIONS IN REPETITIVE CATEGORY           !
!           !           !           !           !           !
! INT    ! NUMBER OF INPUT FIELDS IN SCREEN                     !
!           !           !           !           !           !
! IER    ! NUMBER OF SCREEN-RELATED ERROR MESSAGES           !
!           !           !           !           !           !
! SESSI  ! SESSION NUMBER OF GENERATED PROGRAM               !
!           !           !           !           !           !
! LIBRA  ! LIBRARY CODE                                         !
!           !           !           !           !           !
! USERCO ! USER CODE                                           !
!           !           !           !           !           !
! DATGN  ! DATE OF GENERATED PROGRAM                           !
!           !           !           !           !           !
! TIMGN  ! TIME OF GENERATED PROGRAM                           !
!           !           !           !           !           !
! PROGR  ! PROGRAM CODE                                         !
!           !           !           !           !           !
! PROGE  ! PROGRAM EXTERNAL NAME                               !
!           !           !           !           !           !
! PRDOC  ! HELP PROGRAM EXTERNAL NAME                           !
!           !           !           !           !           !
+-----+

+-----+
!           CHART OF ON-LINE CONSTANTS AND VARIABLES (CONT'D) !
+-----+
!           !           !           !           !           !
! DATOR  ! YEAR-MONTH-DAY FORMATTED MACHINE DATE             !
!           !           !           !           !           !
! DATSEP ! SEPARATOR USED IN DATES                             !
!           ! DEFAULT VALUE: '/'                               !
!           !           !           !           !           !
! DAT6   ! DATE FORMATTING: DDMYY OR YYMDD                    !
!           !           !           !           !           !

```

```

! DAT7 ! ALSO OUTPUT FORMATS (DD/MM/YY FOR INSTANCE) IF !
! DAT8 ! A VARIABLE DATA ELEMENT (V) HAS A DATE FORMAT !
! ! !
! DATCTY ! FIELD FOR CENTURY LOAD !
! ! !
! DAT6C ! NON-FORMATTED DATE WITH CENTURY !
! DAT7C ! ! !
! ! !
! DAT8C ! FORMATTED DATE WITH CENTURY: MM/DD/CCYY !
! ! !
! DAT8G ! GREGORIAN FORMATTED DATE: CCYY/MM/DD !
! ! !
! TIMCO ! TIME !
! ! !
! TIMDAY ! FORMATTED TIME: HH:MM:SS !
! ! !
! 5-scrn-! THIS FIELD CONTAINS THE NAME OF THE !
! PROGE ! PROGRAM TO BRANCH TO !
! ! !
+-----+

```

```

+-----+
!           CHART OF VALIDATION VARIABLES AND INDICATORS           !
+-----+

```

```

! ! !
! ICF ! CONFIGURATION VARIABLE !
! ! '1' = SCREEN IN INPUT !
! ! '0' = NO SCREEN IN INPUT !
! ! !
! OCF ! CONFIGURATION VARIABLE !
! ! '1' = SCREEN IN OUTPUT !
! ! '0' = NO SCREEN IN OUTPUT !
! ! !
! OPER ! OPERATION CODE !
! ! 'A' = INQUIRY !
! ! 'M' = UPDATE !
! ! 'S' = SCREEN CONTINUATION !
! ! 'E' = CONVERSATION END !
! ! 'P' = PREVIOUS DISPLAY !
! ! 'O' = TRANSFER TO ANOTHER SCREEN !
! ! !
! OPERD ! OPERATION CODE FOR DEFERRED BRANCHING !
! ! '0' = DEFERRED CALL OF ANOTHER SCREEN !
! ! INITIALIZED IN F0520 AND MOVED INTO OPER IN F40 !
! ! !
! CATX ! CATEGORY BEING PROCESSED !
! ! '0' = BEGINNING OF RECEPTION OR DISPLAY !
! ! ' ' = SCREEN TOP !
! ! 'R' = REPETITIVE CATEGORY !
! ! 'Z' = SCREEN BOTTOM !
! ! !
! CATM ! TRANSACTION CODE !
! ! 'C' = CREATION !
! ! 'M' = MODIFICATION !
! ! 'A' = DELETION !
! ! 'X' = IMPLICIT UPDATE !
! ! !
! ICATR ! INDICATOR OF CATEGORY BEING PROCESSED !
! ! (REPETITIVE CATEGORY ONLY) !
! ! !
! FT ! END OF REPETITIVE CATEGORY INDICATOR !
! ! '0' LINES TO DISPLAY !
! ! '1' NO MORE LINES TO DISPLAY !
! ! !
! ddss-CF ! SEGMENT CONFIGURATION INDICATOR (seg. ddss) !
! ! '1' THE SEGMENT IS PROCESSED !
! ! !

```

```

!           ! '0' THE SEGMENT IS NOT PROCESSED           !
!           !                                           !
+-----+
+-----+
!           CHART OF VALIDATION VARIABLES AND INDICATORS (CONT'D) !
+-----+
! IK       ! PHYSICAL FILE ACCESS ERROR INDICATOR           !
!           ! '0' NO ERROR                                       !
!           ! '1' ERROR                                         !
!           !                                           !
+-----+
+-----+
!           ERROR VARIABLES                               !
+-----+
!           !                                           !
! SCR-ER ! STORAGE OF SCREEN ERROR                               !
!           ! '1' NO ERROR                                       !
!           ! '4' ERROR                                         !
!           !                                           !
! CAT-ER ! STORAGE OF ERROR ON CURRENT CATEGORY             !
!           ! ' ' NO ERROR                                       !
!           ! 'E' ERROR                                         !
!           !                                           !
!ER-scrn-! MEMORIZATION OF DATA ELEMENT ERROR               !
! delcod ! '0' DATA ELEMENT ABSENT                           !
!           ! '1' DATA ELEMENT PRESENT                       !
!           ! '2' INVALID ABSENCE                             !
!           ! '4' INVALID CLASS                               !
!           ! '5' INVALID VALUE                               !
!           !                                           !
+-----+

```