

VisualAge Pachbase



Business Logic & TUI Clients

Version 3.0



VisualAge Pachbase



Business Logic & TUI Clients

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/productinfo.htm>

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

First Edition (August 2002)

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 1983,2002. All rights reserved.

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

Contents

Notices	v	Call of Segments in the TUI Client Component (-CS)	106
Trademarks.	vii	TUI Client Dialogue Complement (-O)	114
Chapter 1. Foreword	1	Comments (-GC)	118
Chapter 2. Logical View	3	Generated Constants/ Generation of a PSB (-GG)	121
Definition of a Logical View	3	Setting Generation Options (-GO)	125
Description of a Logical View.	5	Writing Procedural Code (-P)	132
Chapter 3. Business Component.	11	Introduction	132
Definition of a Business Component	11	Operators Used by Pacbench Client/Server	132
Call of Segments in the Business Component (-CS)	19	Inserting and Replacing Automatic Sub-Functions	136
Introduction	19	Chapter 6. Monitors.	139
Description & Call of the Error Message File/Table	20	TUI Applications Monitors / Sub-Monitors	139
Screen Description	21	Graphic Application : Communications Monitor	140
Dialogue Complement (-O)	28	Chapter 7. Error Messages - Help Function (TUI)	149
Comments (-GC)	29	Error Messages: Introduction	149
Generation Options (-GO)	32	Business Component:Overriding Automatic Error M.	150
Writing Procedural Code (-P)	34	TUI Client: Overriding Automatic Error Messages	151
Operators Used by Pacbench Client/Server.	34	User-Defined Error Messages	153
Inserting and Replacing Automatic Sub-Functions	39	Help Function (TUI Applications)	154
Inserting a User Service Processing	44	Help Function: Introduction	154
DL/1 Secondary Index Management	44	Help Messages : Coding.	155
Chapter 4. Folder / Folder View	47	Coding Screen (-GE)	157
Definition of a Folder / Folder View	47	Error Messages & On-Line Help	
Description of a Folder / Folder View	54	Generation/Printing	161
Comments (-GC)	57	Example and Limitations	162
Folder (Dialogue) Generation Options (-GO)	60	Description of the Error Message File	164
Chapter 5. TUI Client.	65	Chapter 8. Access Mode.	167
Introduction to TUI Applications	65	Screen : Choices	167
Definition of a TUI Client (Dialogue).	66	Screen Cross-References	168
Call of Data Elements (-CE)	82	Screen Generation and/or Printing	169
Dialogue Simulation (-SIM).	100		

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. Foreword

This manual is the Pacbench Client/Server Reference Manual for the development of Business Logic and TUI Clients (Textual User Interface, also referred to as character-mode interface or dumb client).

It describes each screen and input field as they appear in the VisualAge Pacbase Textual User interface.

PREREQUISITE READING

>>>> Before reading this manual, you should be familiar with the major principles used in the development of Client / Server applications.

These principles are documented in the 'Concepts and Architectures' manual in the Developer's Documentation dedicated to eBusiness Applications. This manual is the entry point to the Pacbench C/S facility.

It is also recommended to read the 'Pacbench C/S - Business Logic' manual which presents the development of a Server component, step by step.

PREREQUISITE WORKING KNOWLEDGE

For complete information, you may refer to the following manuals:

- .The Data Dictionary manual (*),
- .The Structured Code manual (*),
- .In case of data storing in a DBMS:
 - Database manual (*) specific to the DBMS in use.

(*) The contents of these manuals are available as on-line help in the VisualAge Pacbase WorkStation.

.To use the VisualAge Pacbase WorkStation (as it is recommended), refer to the 'WorkStation User Interface' guide which documents the windows graphic interface (except the Folder's or Folder Views, which are only documented in the 'Business Logic' manual).

- Server Component refers to the set of components to define in order to ensure Business Logic, i.e. :

FUNCTIONAL NAME	ENTITY
Business Component (elem.)	Bus. Comp. (0 entity, S type)
I/T Business Component	Bus. Comp. (0 entity, IT type)
Error message Server	Bus. Comp. (0 entity, E type)
Folder	Folder (0 entity, F type)
Folder View	Folder View (0 entity, FV type)
Communications Monitor	Bus. Comp. (0 entity, FM type)
Comm. Mon. simple mode	Bus. Comp. (0 entity, MV type)

- Business Component refers to the set of components which make up the Client part, i.e.:

FUNCTIONAL NAME	ENTITY
TUI Client	C/S screen (0 entity, c type)
TUI Client Monitor	C/S screen (0 entity, MC type)

TUI Client sub-Monitor C/S screen (0 entity, SC type)

>>>> When you use Pacbench Client/Server to develop a graphic application, refer to the 'Graphic Presentation' manual.

Chapter 2. Logical View

Definition of a Logical View

Before defining a Logical View, you must define the Data Structure which it will be attached to ('V-type' Data Structure).

The Definition of a Logical View is documented in details in the 'Pacbench C/S - Business Logic' manual.

```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY                               *PDMC.VDOC.ACC.3!
!
!                               1 2
! LOGICAL VIEW CODE              VU03
!                               3
! NAME.....: PILOT LOGICAL VIEW
!
! NUMBER OF OCCURRENCES.....: 4
!
!
!
! TRANSFER DIRECTION.....: 5
!
!
! EXPLICIT KEYWORDS...:
!
!
! UPDATED BY.....:                ON :                AT:  :  :                LIB : ACC
! SESSION NUMBER.....: 0378        LIBRARY.....: ACC    LOCK.....:
!
! O: C1 CH: S VU03                ACTION:
+-----+

```

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	2	DATA STRUCTURE CODE
		This code is made up of two alphanumeric characters. This is a logical code internal to the Database and therefore independent of the names used in Database Blocks and Programs.
2	2	RECORD CODE
		The first character must be a number. The second one can be a number or letter. If the second one is a letter, the first character must be different from zero.
3	36	SEGMENT NAME
		This name must be as explicit as possible because it is used in the automatic building of keywords, Words used here become implicit keywords (subject to limitations specified in the Character-Mode User Interface Guide, chapter 'Search for Instances', subchapter 'Searching by Keywords').
4	4	NUMBER OF LOGICAL VIEW OCCURRENCES
		The value entered in this field specifies the read or update iterative capacity of a server that calls this Logical View. This capacity is expressed by a maximum number of repetitions. The Logical View can then be used as a repeated structure.
		NOTE: If the Logical View is not a repeated structure
		as a whole and if only a few of the Data Elements which constitute it belong to a repeated structure, you must specify it in the DATA TYPE field, in the -CE screen (for more explanations, refer to it).
		CAUTION: this does not apply to a graphic application.
	999	Maximum authorized value.
5	1	DIRECTION OF CLIENT-SERVER TRANSFER

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			In a Logical View which is a repeated structure (previous field specified), it represents the transfer direction between the client and the server:
		C	Client to server
		S	Server to client
			NOTE: You only input 'C' or 'S' if all the Data Elements have the same transfer direction. Otherwise specify it in the -CE screen of the Logical View.
		blank	in both directions. This value is required for a Logical View used in a graphic application.
6	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'.

Description of a Logical View

The Description of a Logical View is documented in the 'Pacbench C/S - Business Logic' manual.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
5	6		DATA ELEMENT CODE
			ELEMENTARY DATA ELEMENT DEFINED IN THE REPOSITORY
			The Data Element necessarily uses the characteristics defined at the Repository level. If it is used as a group, its format depends on its contents. If the group is used as an identifier, its contents must be compatible with the Repository format.
			NOTE: Data Elements with an A(n) or X(n)9(n) format are not authorized in a graphic application.
			DATA ELEMENT NOT DEFINED IN THE REPOSITORY
			The characteristics must be indicated at the Segment level.
			RESERVED OR FORBIDDEN CODES
		FILLER	The 'SUITE' Data Element is not allowed, it is used by the system during Program generation. Data Element which we do not anticipate using. NOTE: This type of Data Element can not be used in a graphic application.
		FILLER	Do not use this Data Element (alignment fields). Not authorized for the CS/INTERFACE.
			CALL AGGREGATE DATA
			You can indicate a Segment code.
			The GROUP field, in this case, has a specific call codification. The called structure is inserted in the Segment description where it is called, it cannot be changed or completed. All information belonging to the called structure is taken into account. The call can be done from inside a repeated group.
			For a C2 display, the Action code indicates the level of nesting (display of up to 4 levels of nesting).
			CONTINUATION LINES
			It is possible to use continuation lines.
			In this case, do not indicate the Data Element code.
			It is necessary that the line number follows the line number containing the Data Element code.
6	18		NAME OF DATA ELEMENT
			It is not required for a Data Element which is not defined in the Data Dictionary.
			However, it is optional for a data aggregate or a FILLER.
			NOTE: For on-line entry of Data Elements that are not declared in the Dictionary, this field cannot be used to input more than one Data Element at a time. There is actually only one available field on this screen, whether for input or for display.
			To define an Element at the Segment level :
			- Enter the Element code (and possibly the format) on the -CE, line nnn,

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		- On the 'name' line, repeat the line number (nnn), and indicate the name (18 characters maximum),
		- Use the C2 option to view the name and format.
		NOTE: If several undefined Data Elements have been defined in the Dictionary, only the name of the first Data Element will be displayed if the Choice 'CH:S.....CE' is used.
		To view the name of the Data Element CODEL, on line 130, for example, use the choice 'O: C2 CH: Sssss-CE130'. This will display the Data Elements called in the Segment 'ssss' from the line 130 on.
7	10	INTERNAL FORMAT OF THE DATA ELEMENT
		This format is required for:
		- the Elementary Data Element not defined in the Dictionary, created at the Segment level (COBOL format),
		- the FILLER Data Element.
		This is the internal format, defined in the same way as the Data Element definition. The input and printing format will be identical, but the use will be DISPLAY.
8	1	INTERNAL USE
		For Data Elements not defined in the Specifications Dictionary when the INTERNAL FORMAT OF DATA ELEMENT field has been given a value, enter the appropriate USAGE (default : 'D' for DISPLAY).
		For valid values, see the USAGE field on the Data Element Definition Screen.
9	1	DATA TYPE
	U	Identifier for the pagination of a list for the Client. As output of the server, it contains the selection criteria values of the next page.
	R	Repeated structure (TUI application only): There is only one repeated structure for one Logical View. A repeated structure is specified in the group Data Elements. This Data Element indicates the beginning of the repeated Data Elements. The end of the repeated Data Elements is deduced from the number of Data Elements in a group. The number of occurrences must be indicated in the NUMBER OF REPETITIONS field. The number of elementary Data Elements must be indicated in the NUMBER OF DATA ELEMENTS field.
		NOTE: if the Logical View as a whole is a repeated structure, you must specify it in the NUMBER OF OCCURRENCES field in the Definition of the Logical View (refer to the documentation of this field).
	E	Extraction parameter : The Data Element is used as a parameter for an extraction method.
		The 'E' type Data Elements must be entered in the '-CE' screen before the Data Elements that describe the Logical View's contents.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Each identifier Data Element ('U' type) is an implicit extraction parameter. The other Data Elements describing the Logical View cannot be used as parameters. If necessary, define and call a child Data Element.
10	2	NUMBER OF DATA ELEMENTS IN A GROUP
		PSEUDO-NUMERIC FIELD
	1-99	A group is defined by its number of Elementary Data Elements (the Segment call corresponds to an elementary Data Element).
		The groups can contain up to 99 elementary Data Elements.
		9-level groups are allowed. They cannot overlap.
		It is possible to use this field in order to call the MERISE entities or Segments that are already defined.
		Their codes will be indicated in the Data Element code field (aggregate data call).
	**	Call of Segments.
		NOTE: The group Data Elements are not taken into account in the graphic application.
11	3	NUMBER OF REPETITIONS
		PURE NUMERIC FIELD
		OCCURS clause, at the elementary Data Element level.
		It is not possible to have a repeated group area apart from the repeated structure (see the DATA TYPE field).
		Graphic applications: The repeated Data Elements are not authorized.
12	1	DATA ELEMENT PRESENCE
	O	Required. Its presence check is generated in the Client, and in the Business Component if the option CHECKSER=YES.
		NOTE: This generation includes the checks in relation to the Data Element Description.
	P	Required. Its presence check is generated only in the Business Component, if the option CHECKSER=YES.
		NOTE: This generation includes the checks in relation to the Data Element Description.
	blank	Optional. Only the checks in relation to the Data Element Description are generated in the Client, and in the Business Component, if CHECKSER=YES.
	F	Optional. The checks in relation to the Data Element Description are only generated in the Business Components if CHECKSER=YES.
13	1	DIRECTION OF CLIENT-SERVER TRANSFER
	C	Client to server
	S	Server to client
	BLANK	In both directions.
		This value is required for a Data Element or a Segment used in a graphic application.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		The direction of the transfer must also be indicated in the group Data Elements.
		The repeated Data Elements must have the same transfer direction, which can only be indicated on the Data Elements bordering the repeated data (Data Element with 'R' indicator).
14	1	TRANSFER OPERATOR
	+	Transfer by addition.
	-	Transfer by subtraction.
	M or BLANK	Transfer by substitution. Note: A transfer of this type can be done from a group Data Element. In the Server --> Client direction, this type of transfer is systematically generated.
15	10	SUB-SCHEMA ASSIGNMENT
		This field defines one or more sub-schemas for a Logical View. This concept is used on a Logical View called in any type of a Folder's node when developing a graphic application in folder mode.
		To specify that a Data Element belongs to a n sub-schema, enter an O in the column number n in the SUB-SCHEMA area (10 columns).
		NOTE: The sub-schemas appear in the first call line of each Data Element. For a batch update of this field, the O(s) must be entered in uppercase.
16	10	TRANSFER AREA
		This field is made up of the Segment, Screen and Data Element code updated by the Logical View data or transferred after selection of the data from the Logical View, depending on the transfer direction set in the DIR column.
	R*	Allows you to redefine a Data Element in a Logical View. The Data Element redefines the first preceding Data Element with the same level.
		NOTE: The check indicator is not generated for a redefined Data Element (CH-vvnn-corub). The Logical View transfer to the redefined Segment zones is therefore not conditioned by the presence vector.
		>>>> The specification of redefined Data Elements recovers from a particular use and is only recommended when reusing an existing one.
17	1	DOCUMENTATION INDICATOR
		This field is used in on-line mode only. It is a read-only field.
	*	A Comment, a Generation Element or an Error Message has been assigned to the element called on this line.
		Access to line nnn: -CEnnn, or -Dxnnn for a Database Block (with x = C, H or R depending on the Block type)
		To access the Comment, Generation Element or Error Message assigned to the called element, enter the access to line nnn followed (without blank) by GC (for Comment), GG (for Generation Element) or GE (for Error Message).

Chapter 3. Business Component

Definition of a Business Component

The Definition of a Business Component is documented in details in the 'Pacbench C/S - Business Logic' manual.

```

+-----+
!           ENGLISH DOCUMENTATION LIBRARY           PDSG.VDOC.ACI.439  !
!           1 2                                     !
! ON-LINE DIALOGUE CODE           ZS                 !
! DIALOGUE NAME.....: 3 FLIGHT RESERVATION (SERVER) !
! SCREEN TYPE.....: 4 S           BUSINESS COMPONENT !
!                                     !
!                                     !
!                                     !
!                                     !
! VARIANTS.....:5 X 6 0   IBM VS2 CICS (PROG.MAP BMS) !
! CONTROL CARD OPTIONS FRONT & BACK..: 7-8   (PROGRAM) 9 $$ 10   (MAP)!
! EXTERNAL NAMES .....: 11   (PROGRAM) 12   (MAP)!
! TRANSACTION.....:13                                     !
!                                     !
! EXPLICIT KEYWORDS...: 14                               !
!                                     !
! UPDATED BY.....:           ON :           AT: : : LIB : ACI   !
! SESSION NUMBER.....: 0383   LIBRARY.....: ACI   LOCK.....:   !
! O: C1 CH: 0 zs           ACTION:                       !
+-----+

```

```

+-----+
!               ENGLISH DOCUMENTATION LIBRARY           PDSG.VDOC.ACC.439  !
!               1 2                                     !
! ON-LINE SCREEN CODE           ZS77PI                   !
! SCREEN NAME.....: 3 PILOT INITIALIZATION (SERVER)    !
! SCREEN TYPE.....: 4 S           SERVER COMPONENT      !
!
!
!
! VARIANTS.....:5 X 6 0      IBM VS2 CICS              !
! CONTROL CARD OPTIONS FRONT & BACK..:7 8      (PROGRAM) 9 10      (DIV)!
! EXTERNAL NAMES .....:11 WSSVPIP (PROGRAM) 12      (DIV)!
! TRANSACTION.....:13
!
! EXPLICIT KEYWORDS...: 14
! UPDATED BY.....:          ON :           AT: : :      LIB : ACC  !
! SESSION NUMBER.....:0377   LIBRARY.....: ACC   LOCK.....:
!
! 0: C1 CH: 0 zs77pi           ACTION:
+-----+

```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			COMPLETE CODE
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	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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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.
5	1		TYPE OF COBOL TO GENERATE
		N	No generation
			ON-LINE SYSTEMS DEVELOPMENT FUNCTION:
		X	IBM MVS COBOL II
		O	IBM MVS COBOL/VS
		X	IBM VSE COBOL II
		1	IBM VSE COBOL/VS
		3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT COBOL
		O	IBM AS/400 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 OR VSE COBOL II

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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).
		3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT 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
6	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
		0	Variants 0, 1, X for CICS (IBM) (programs and BMS maps).
			Variant 2 for IBM 36, monochrome map.
			Variant 3 for PC/MICROFOCUS (MS/DOS).
			Variants 4, 5, 6 for QUESTAR.
			Variant I for DEC/VAX.
			Variant U for UNISYS 2200 (programs and FLDP maps).
			Variant 8 for UNISYS Series A SDF format.
			Variant Y for IBM 38.
			Variant O for AS 400.
			Variant M for GCOS6-DTF-DFC-Vision.
			Variant 7 for VPLUS.
		1	Variants 0, X for IMS (IBM) (programs and MFS maps).
			IMS (IBM) (programs and MFS maps).
			Variant 2 for IBM 36, map in color.
			Variant 3 for PC/MICROFOCUS (OS2).
			Variants 4, 5, 6 for VIP TYPE.
		2	Variants 0, 1, X for CICS (IBM), BMS map in color.
			Variant 3 for MICROFOCUS (UNIX).
		3	Variants 0, X for IMS (IBM), MFS map in color.
		4	Variants 0, 1, X for IMS (IBM) monitor.
			Variant 3 for IBM VISUALAGE COBOL program.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	5	Variants 0, 1, X for CICS (IBM) monitor.
	C	Variants 4, 5, 6, U for MULTI-TERMINAL.
		Variants 0, 1, X for CICS MULTI-TERMINAL.
		Variant K for ICL.
		Variant 8 for UNISYS Series A Logical Screen.
	F	Variant 4 for TDS FORMS (BULL GCOS7).
		Variant M for DM6 TP FORMS.
		Variant R for 'TPSVRINIT service' type Program or 'TPSVRDONE service' type Program.
	R	Variant F for 'REQUESTER' type Program. or variant R for 'CLIENT' type program.
	S	Variant F for 'SERVER' type program. or variant R for 'SERVICE' type program.
		PACBENCH CLIENT/SERVER
		TP MONITOR
	0	Variants 0, X for CICS (IBM) (Program and BMS map for the client).
		Variant 3 for MICROFOCUS MS/DOS.
		Variant F for TANDEM Pathway.
		Variant R for TUXEDO.
		Variant 7 for VPLUS.
		Variant U for UNISYS-2200.
		Variant 8 for UNISYS-A.
	1	Variants 0, X for IMS.
		(program and MFS map for the client).
		Variant 3 for MICROFOCUS OS/2.
		Variant F for TANDEM TUXEDO.
		Variant 8 for UNISYS-A Open/OLTP.
	C	Variants 0, 1 for MULTI-SCREEN CICS or variant X for CLIENT ONLY.
		Variant 4 for MULTI-SCREEN GCOS7.
		Variant 5 for MULTI-SCREEN GCOS8.
		Variant R for TUXEDO (from release 6.2 and TUXEDO (from release 6.2 and higher).
	2	Variants 0, X for CICS (IBM) (program and BMS color map).
		Variant 3 for MICROFOCUS UNIX.
	3	Variants 0, X for IMS (program and MFS color map).
		Variant 3 for MICROFOCUS.
	4	Variant 3 for IBM VISUALAGE COBOL.
7	1	CONTROL CARDS BEFORE PROGRAMS
		Option code which selects the JCL lines to be inserted before each generated program.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	\$	No generation of program.
		Business Component / single-view (with no specification of a Folder). The value '\$' is used to disable the Business Component generation and to enable the Services Manager generation.
8	1	CONTROL CARDS AFTER PROGRAMS
		Option code which selects the JCL lines to be inserted after each generated program.
	\$	No generation of program.
9	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 '\$' is used to disable the generation of the Services Manager and to enable the Business Component to be generated.
10	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.
11	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.
12	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).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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 and Architectures' manual.
13	8		TRANSACTION CODE
			ON-LINE SYSTEMS DEVELOPMENT:
			CICS: 4-character transaction code (default).
			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.
			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.
14	55		EXPLICIT KEYWORDS
			This field allows you to enter additional (explicit) keywords. By default, keywords are generated from the instance's name (implicit keywords).
			Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage and are therefore ignored in keywords.
			Keywords are not case-sensitive: uppercase and lower-case letters are equivalent.
			NOTE: 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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			For more details, refer to the 'Character Mode User Interface' guide, chapter 'Search for Instances', subchapter 'Searching by Keywords'.

Call of Segments in the Business Component (-CS)

Introduction

For each Business Component, specify the Logical view you are using, and, for this view, indicate the accessed Segments of a file, a table or a SQL view.

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

NOTE: You can declare several Logical Views, although it is not recommended. In this case, the total number of description lines associated with these Logical Views is limited to 100.

Segments, having the same Data Structure code, must have the same organization, the same description type and, for V-type organization, the same external name.

GENERAL CHARACTERISTICS

For each Logical View, the organization only is required. Its code must be the same as that used for the corresponding Logical View called in the associated TUI Client(s).

For each Segment, indicate:

- The processing category (before repetitive, repetitive, and after repetitive).
- The type of logical access processing (validation, update and selection) to perform on the Segment and the key filling mode.
- The use in the generated processing.
- Whether access is conditioned by access to another Segment.
- The physical organization.

NOTE: It is recommended to use the Pacbench function of the VisualAge Pacbase WorkStation to specify a Business Component's Segment calls.

>>>>& For additional information, more suited to the graphic interface, refer to the 'Pacbench C/S - Business Logic' manual and to the 'WorkStation User Interface' guide.

NOTES FOR 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).

For the generation of access procedures, 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.

Description & Call of the Error Message File/Table

For a graphic application, you must define an error message server - E type Business Component - to access the error message file or table.

NOTE: For a TUI application, you can do the same or call the error message file or table at the level of your application C/S Screen Dialogue. In this case, you do not have to define and describe this file or table.

This file (or this table) is described by a Segment, called in the -CS screen of the error message server.

The code of this dedicated Segment has to be EM00 in the Business Component.

However, its code in the VA Pac Database is the user's choice (ER00 and ER10 in the following examples).

INDEXED OR WORKING TYPE ORGANIZATION :

- The description of the Segment must be entered as follows :

```
SEGMENT DESCRIPTION : EM00 ERROR MESSAGE
+-----+
!A NLG : CORUB FORM.INT. U OCC GR I CMS456 etc.!
! 100 : ERKEY                10                !
! 110 : APPLI XXX           D                   !
! 120 : TYPEN X             D                   !
! 130 : LNG X               D                   !
! 140 : XCLEF                6                 !
! 150 : PROGR X(6)          D                   !
! 160 : CVIEW XXXX          D                   !
! 170 : DELEM X(6)          D                   !
! 180 : TYCOD X             D                   !
! 190 : ERCODE              2                 !
! 200 : TYERR X             D                   !
! 210 : NUERR XXX           D                   !
! 220 : NULIG 999           D                   !
! 230 : GRAER X             D                   !
! 240 : LIERR X(66)         D                   !
! 250 : FILLER (4)         D                   !
+-----+
```

- The call in the -CS of the error message server must be the following :

```
ON-LINE SCREEN CALL OF SEGMENT
-----
C SEGM      : T USE PREC ACCESS KEY  R  T NAME  SEGM N
A CODE C NL : G R D SEGM SOURCE  KEY  A O D EXT.  LIB S
  EM00  00 :                ERKEY  V 1  DGCSLE ER00
-----
```

The code of the external name corresponds to the file logical name (DDNAME in MVS).

RELATIONAL TABLE :

- For a SQL/DS table, the Segment description must be the following :

SEGMENT DESCRIPTION : ER10 SQL ERROR MESSAGE

A NLG : CORUB FORM.INT. U OCC GR I CMS456
100 : ERKEY X(29) D
110 : LIBELE X(71) D

- The call must be the following :

ON-LINE SCREEN CALL OF SEGMENT

C SEGM : T USE ACCESS KEY R T NAME SEGM N
A CODE C NL : G R A SEGM SOURCE KEY A O D EXT. LIB S
EM00 00 : ERKEY H 1 SQLBLC ER10

The code of the external name must contain the Block code in which the ER10 table description has been included.

OTHER TYPES OF DBMS :

- The Segment must have two Data Elements with the same length as above but with different codes.
- The call is made the same way, the key code corresponding to the code of the Segment first Data Element.

Screen Description

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			The Logical View code ('S' organization) must be identical to that used in the corresponding Logical View called in the associated TUI Client(s).
			At least one Logical View must be present in a S-type Business Component.
4	1		CATEGORY INDICATOR
			Used to indicate the processing category to which this Segment belongs.
			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 category.
			NOTE: value required for a graphic application.
		Z	Segment belonging to the screen-bottom.
			NOTE: do not enter a value for this field on the Logical View call line.
5	2	NUMER.	LINE NUMBER
			PURE NUMERIC FIELD
		00	Only one '00' line number per Segment is permitted. (Default)
			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	No generation of logical access processing (Organization different from S).
		P	No description of Segment. (Organization different from S).
7	1		USE IN VALIDATION/UPDATE
			Used to specify the way in which the Segment is used in validation and update processing.
			Organization different from S, X, and 2:
		N BLANK	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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			A read command with the setting of an error code is generated (if it is not found). If on the same line the selection use is 'A' and if the validation is correct, a selection is performed.
		M	Updated Segment (modification only).
		S	Updated Segment (deletion only).
		C	Updated Segment (creation only).
		X	Updated Segment (creation and modification only).
		T	Updated Segment (creation, modification and deletion).
			In the last five cases, the existence validations and the update statements corresponding to the chosen options are generated.
			For S organization, the BLANK value is forced.
			For organization = X or 2:
		N BLANK	No server call is generated.
		L	Server call for selection.
		E	Server call for validation.
		M	Server call for validation and update.
		T	Server call for validation, update and selection.
		X	Server call for validation and selection.
		U	Call of a User Service.
8	1		USE IN SELECTION
			For organizations different from S, X or 2 :
		A	Used in selection.
			The access order will be generated for the selection according to the type of logical access processing.
		BLANK	No order is generated for this Segment in the selection processing (default option).
			For organization = S:
		N BLANK	BLANK is the default value.
			For organization = X or 2 :
		N BLANK	No server call is generated.
		A	Server call for selection (not compatible with the 'L' value in reception).
		U	Call of a User Service
9	4		PREVIOUS SEGMENT CODE
			It is possible, in each category, to access several Segments.
			Therefore, a hierarchy can be indicated concerning the access to the Segment of a category. This field contains the previous Segment code in the access hierarchy. This Segment must belong to the same category.
			If several Segments have the same previous Segment, they are processed in alphabetical order of their code.
			Each Segment must be linked directly or indirectly to the Logical View on which it depends. It must include the code of a previous Segment or of its Logical View (S organization).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Exceptions: The error message Segment must not include the previous Segment. A Logical View cannot have a previous Segment.
			NOTE: It is not recommended to link Server calls and Segment calls that correspond to databases or files.
10	14		ACCESS KEY SOURCE
			This field can be a working area or a literal.
			For example: '02 ' for a literal, ww10-delco for a working area.
			NOTE: The key loading for the lists must always be performed from the 1- area in the servers if the key is in the repetitive of the Logical View.
11	6		ACCESS KEY (DATA ELEMENT CODE)
			Organization different from S only:
			This is the DATA ELEMENT CODE used as the access key to a Segment. For a VSAM file (V ORGANIZATION), it must be entered on the segment's '00' line. The Data Element entered 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), the selection of a sub-system is executed by indicating the value 'NUSSY' in this field. The system generates the associated key (G-ddss-NUSSY).
			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 PSB).
			SQL Database:
			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.
			ERROR MESSAGE FILE OR TABLE
			If the error message Segment is a SQL/DS Segment or if it has a V or W organization, the Data Element key must be ERKEY.
12	1		CONTROL BREAK INDICATOR FOR SELECT
			For organization different from S only:
			Indicates that a Data Element, belonging to a key in the repetitive part, must remain constant during the selection.
		BLANK	No selection control break.
		C	Selection break.
			SQL ORGANIZATIONS:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		E	Selection control break; causes the generation of an equal condition in the WHERE clause of the SQL 'DECLARE CURSOR' statement.
		R	H, D, or V ORGANIZATIONS:
			Large reading of the Folder. Field required for each Segments associated with the Logical View, this View must belong to a dependent node of the Folder. The value must be set on each key Data Element corresponding to a key Data Element of the Logical View that belongs to the Folder's root node.
13	1		ORGANIZATION (REQUIRED IN CREAT)
			Required for the '00' LINE NUMBER of the Segment.
		V	Segment Organization: Indexed.
		G	'TABLE'-type Segment. In this case, the generated access corresponds to the call of the Pactables general module.
		H	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. The type of the relational database Block is required on the Block definition indicated in the EXTERNAL NAME field.
		D	DL/1 Segment
		A	ADABAS Segment
		A	Only one description in WORKING is generated.
		W	Segment accesses managed by specific procedures. Only one description in WORKING is generated. In the -CS Screen of the Server Dialogue, call the user buffer Segment.
		S	Call of the Logical View.
		X	Call of a Business Component All the Segments that are called in the repetitive and in selection must be of the same type: you cannot have Segments with 'X' organization with Segments with other organizations.
		2	Call of a Business Component instance by instance
			NOTE FOR THE ERROR MESSAGE FILE :
			The error message file can be attributed the following organizations :
			V : its external name and its code in the VA Pac Database are not validated.
			W : WORKING - no access generated.
			H : SQL - the external name corresponds to the Block code (containing the relational database) indicated in the SEGMENT CODE IN THE DATABASE field. The description and the accesses will be generated under the name corresponding to the Library code).
14	1		DESCRIPTION TYPE
		blank	This field allows you to obtain from the Segment description in the database, various descriptions in the generated program. Complete Segment (common and specific parts).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	1	Specific part only. (Forced value for Logical View, DL1 Segment or SQL table).
	E	TABLE-type Segments: The input format of the Segment's Data Elements is taken into account.
	2	Complete segment with variable length for indexed organization (V).
		NOTE: The generated description may vary according to the generation variant.
	3	For a TUXEDO Business Component of SI or ST type, this value associated to a SQL Sybase XA block calls the TUXEDO TPOPEN or TPCLOSE routine to handle connect and disconnect orders respectively.
		NOTE: When this value is not specified, the CONNECT/DISCONNECT orders are automatically generated.
15	8	EXTERNAL NAME OF THE FILE
		Default value : first two characters of the Segment code in the Screen.
		V ORGANIZATION (indexed):
		IBM CICS-VSAM: DDNAME (1 to 8 positions).
		GCOS8 - TP8 :
		- if blocking by characters : external name on 4 characters (generated in the SELECT clause) and blocking on the last four characters (generated in the FD clause).
		- if blocking by record : external name on 5 characters and blocking on the last three characters. The two parts of the external name, each on two characters, are separated by a blank.
		Other variant: Logical name, generated in the SELECT clause.
		H ORGANIZATION (SQL):
		Code of the SQL Block in which the Table or View is called. It is possible to select 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).
		G ORGANIZATION (Pactables):
		In CICS: 4 characters for the DDNAMEs of the files managed by Pactables (these are the characters of the transaction code used to access Pactables files).
		In IMS: Any number of characters that build the names of the 2 PCBs associated with the Pactables Tables. The resulting names will appear as follows:
		.xxxxxD: PCB of the description database
		.xxxxxV: PCB of the contents database
		These names cannot exceed 6 characters.
		The PCBs must belong to the Screen's PSB.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		D ORGANIZATION (DL/1):
		PCB code.
		S ORGANIZATION (Logical View):
		Displays by default the Data Structure code.
		X or 2 ORGANIZATION :
		VisualAge Pacbase code of the Business Component.
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-type 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's '-DR'.
	BLANK	All the Data Elements of the Segment.
	1 to 9	Sub-description (or sub-schema) number 1 to 9
	0	Identifies the sub-description (or sub-schema) number 10.
		V-type organizations: Secondary access keys to indexed files:
	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 validation/update or selection, 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
		Not used for a Business Component.

Dialogue Complement (-O)

For a Business component, this screen (CH: O..O) screen allows to indicate the PSB code in case DL1 Databases or PCB Alternate are used (required when CALLTYPE=CHNG option).

PREREQUISITE

The Dialogue must be defined.

```

+-----+
!           ENGLISH DOCUMENTATION LIBRARY           PDMC.VDOC.ACC.439 !
! DIALOGUE COMPLEMENT....: 1 ZS FLIGHT RESERVATION (SERVER) !
! ! !
! COMMON AREA-DATA STRUCTURE CODE.....: !
! ! !
! ERROR MESSAGE FILE CHARACTERISTICS !
!           ORGANIZATION....: !
!           EXTERNAL NAME....: !
! ! !
! FIRST SCREEN CODE OF THE DIALOGUE.....: !
! ! !
! COMPLEMENTARY COMMON AREA LENGTH.....: !
! ! !
! CODE OF PSB OR SUB-SCHEMA.....: 2 !
! ! !
! OPTIONS : !
! ! !
! SESSION NUMBER      : 0395  LIBRARY      : ACI !
! ! !
! 0: C1 CH: 0 zs 0           ACTION: !
+-----+

```

NUM	LEN	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	6		CODE OF PSB OR SUB-SCHEMA
			For the IMS cobol generation variant, this field contains the PSB code of the application in case DL1 Databases or PSB Alternate are used (required when CALLTYPE=CHNG option).
			For the HP3000 cobol generation variant, this field contains the ALLBASE/SQL Database Block code when the connection to the database is performed via the client monitor.

Comments (-GC)

The 'Comments' screen (-GC) enables you to write and edit text in relation to one selected instance. This screen is available on all entities.

Each comments line is made up of a 60-character field containing text and a 'Type of Line' field for particular purposes (comment, call of a P.I.A, indication of an alias).

CALLING A P.I.A.

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

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

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

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

Additional lines cannot be inserted between called lines.

For more information see the 'Character Mode User Interface' guide, chapter 'Documentary Facilities', subchapter 'Parameterized Input Aids'.

PREREQUISITES

The instance must be defined prior to being documented.

The called P.I.As must also exist.

LINES COPY

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

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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
4	60	DESCRIPTION - FIRST PART
		If no P.I.A. is called, you can use the whole field.
		On a blank-type line, you enter a free comment.
		In order to select only certain lines of the comments of an instance, use print option -EG and indicate the \$OFF command at the left end of each line to be ignored. To re-activate the -EG option on a line, indicate a \$ON command, left-justified, after the last line to be ignored.
		On an A-type line, enter the COBOL Segment name. (see chapter 'Generation of Copy Book', subchapter 'Description of VA Pac PIA 'Data'', value 'A*' in the 'Data Structure Code in Gener. Descr.' field.)
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.

Generation Options (-GO)

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

- on an SQL Database Block, you specify options such as the prefixing, the transformation of lowercase into uppercase...
See the 'SQL Databases' Manual.
- on a Segment, you can inhibit the conversion of lowercase letters into uppercase letters on customized SQL accesses (with the UPPER=NO option).
See the 'SQL Databases' Manual.
- on a Program, you can change the function/subfunction number of the date transformation function (F9520 by default), with the DATPRO=ffss option.
See the 'Batch Applications' manual.
- For all entities managed by the Pacbench C/S Function except blank-type or MW-type entities, you can indicate various generation options.
See the 'Business Logic' Volume of the Developer Documentation dedicated to eBusiness applications.

All these Manuals can be consulted at our internet address:

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

This link to the VA Pac Documentation web site requires a specific login with password. If you are not a registered user, send your login request to vapacsup@fr.ibm.com.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
4	60		DESCRIPTION
			If no P.I.A. is called, you can use the whole field.
			On an O-type line, Enter the option followed by an equal sign and the value (with no space).
5	40		DESCRIPTION / SECOND PART
			This field is specific to a P.I.A. call.
			With value 'C2' in the OPERATION CODE field, the cursor automatically tabs to the first position of this field.
			This field is initialized with 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.

Writing Procedural Code (-P)

>>>> For complete information on this subchapter, the following readings are recommended:

the 'Structured Code' manual for details on the syntax used to write specific procedures,

the 'Pacbench C/S - Business Logic' manual for details on the insertion rules in the different services.

Operators Used by Pacbench Client/Server

This section presents a selection of operators often used with Pacbench C/S.

To consult the complete list of all operators, refer to the 'Structured Code' manual, chapter 'Modifying the Procedure Division', subchapter 'Procedural Code Screen' section beginning with 'Operators'.

SEGMENT ACCESS OPERATORS

Xaa: Followed by a Segment code in the OPERAND, generates a PERFORM of label F80-ffnn-aa.

If the Segment code is followed by an extraction method (after a space), generates a PERFORM of F80-ffnn-aa-code of extraction method (Server components only).

Yaa: Generates a COBOL paragraph label for Function 80. Followed by a Segment code in the OPERAND field, generates a F80-ffnn-aa COBOL paragraph label (used when the physical access to a Segment is replaced).

For these operators, 'aa' is a generated code.

The values of this code are documented in the 'Pacbench C/S - Business Logic' manual.

OPERATORS FOR ERROR POSITIONING

Specific errors are coded in Procedural Code lines (-P). The corresponding error messages are entered according to rules specified in subchapter 'User-Defined Error Messages', Chapter 'Error Messages - Help Function'.

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

ERU: User-related error

Operand:

1 to 4: Error code (user-defined)

5: Blank

after 5: For a repeated Data Element, addition of an area (generally ICATR) which contains the number of occurrences of the repetition.

ERR: Error on a Logical View Data Element.

The code of the Logical View is taken from the 05*C function in which this operator is found.

>>>> For more details on the 05*C functions, refer to chapter 'Business component', subchapter 'Inserting and replacing Automatic Sub-Functions'.

Operand:

1: Code of the error (alphanumeric character except for '0' and '1' which are reserved for coding documentary messages)

2: Blank

after 2: Code of the variable Data Element to which the error code is associated.
For a repeated Data Element, addition of an area (generally ICATR) which contains the number of occurrences of the repetition.

These operators for error positioning trigger the generation of a group of lines ending with a period. If one of these operators belongs to a level 99, it must be its last instruction.

ERL: Error on Logical Lock and Unlock.

In a graphic application with a specified Folder or in single-view mode, this operator allows to set an error on a LOCK or an UNLOCK requested by a graphic client, for a locked or unlocked occurrence respectively.

This (un)lock option is documented in the 'Pacbench C/S - Business Logic' manual.

This operator does not use operands.

OPERATORS FOR END-OF-PROCESSING

GF: GO to end of automatic sub-function.

If the OPERAND field does not have a level, the end-of-processing is done with respect to the automatic sub-function that has been inserted by *A or *P.

If the OPERAND field contains a level, the end of processing is done with respect to the automatic sub-function of the corresponding level.

If the OPERAND field contains ENDV, end of Logical View processing (Branch to the Fvunn-ENDV label).

For more details, refer to Subchapter 'Inserting and Replacing Automatic Sub-Functions'.

GFA: End of selection processing

(Branch to the Fvunn-SELC-FN label).

GFR: End of check/update processing

(Branch to the Fvunn-CHUP-FN label).

GDB: Return to the beginning of current iteration

(Branch to the Fvunn-CHUP-CATR-SVRx-CHCK
or Fvunn-SELC-CATR-SVRx-SELC label).

GDI: Go the beginning of next iteration

(Branch to the Fvunn-CHUP-CATR-SVRx-DONE
or Fvunn-SELC-CATR-SVRx-DONE label).

GFT: Go the end of iteration

(Branch to the Fvunn-CHUP-CATR-FN
or Fvunn-SELC-CATR-FN label).

GF ENDV

Go to the end of Logical View processing

(Branch to the Fvunn-ENDV label)

NOTE :: With these operators, the generated instruction always ends by a full stop. No EL type of conditioning should then be used on a line following a branching operator. Indeed, this might trigger a compiling error in the generated COBOL.

OPERATORS FOR PROCESSING CALL

XT: Allows you to call an elementary processing (transfer, processing on Segment, ...).

Y: Allows you to generate the label of the automatic function when it has been replaced by the insertion of specific code.

For more details, refer to subchapter 'Inserting and Replacing Automatic Sub-Functions'.

OPERATORS FOR DATE PROCESSING

AD6: Transforms the system date into a 6-character date format of 'MMDDYY' or 'DDMMYY' depending on the value in the DATE FORMAT IN GENERATED PROGRAMS field specified on the Library Definition screen. The transformed date is moved into the Data Element indicated in the OPERAND field.

AD8: Transforms the system date into an 8-character date with slashes, 'MM/DD/YY' or 'DD/MM/YY' depending on the value in the DATE FORMAT IN GENERATED PROGRAMS field on the Library Definition screen. The transformed date is moved into the Data Element indicated in the OPERAND field.

AD: Transforms the date according to the format types specified in the OPERAND. A date may be formatted in the following ways :

```

+-----+-----+
! CODE ! GENERATED FORMAT !
+-----+-----+
! I   ! 'Internal' : YYMMDD !
!-----!-----!
! D   ! 'Display' : MMDDYY or DDMMYY !
!     ! according to the value entered in the DATE FORMAT !
!     ! IN GENERATED PROGRAMS field on the Library Def. !
!     ! screen. !
!-----!-----!
! E   ! 'Extended' : MM/DD/YY or DD/MM/YY !
!     ! according to the value entered in the DATE FORMAT !
!     ! IN GENERATED PROGRAMS field on the Library Def. !
!     ! screen. !
!-----!-----!
! S   ! 'Internal' : CCYYMMDD !
!-----!-----!
! C   ! 'Display' : DDMMCCYY or MMDDCCYY !
!     ! according to the value entered in the DATE FORMAT !
!     ! IN GENERATED PROGRAMS field on the Library Def. !
!     ! screen. !
!-----!-----!
! M   ! 'Extended' : DD/MM/CCYY or MM/DD/CCYY !
!     ! according to the value entered in the DATE FORMAT !
!     ! IN GENERATED PROGRAMS field on the Library Def. !
!     ! screen. !
+-----+-----+
! G   ! 'Extended' : CCYY/MM/DD !
+-----+-----+

```

The different formats can be set at the Data Element input, internal or output format level or at the conversational format level.

NOTE: When several lines are required to enter the condition of a date processing operator, the operands must be written on the line following the last one and the condition.

EXAMPLE: In order to change an 'T'-formatted date into a 'D'- formatted date, 'AD' should be entered in the OPERATOR field and 'ID DELCO1 DELCO2' in the OPERAND field; DELCO1 being the Data Element containing a YY/MM/DD date format (the Pacbase constant DATOR may be used) and DELCO2 being the Data Element containing the changed date format: DD/MM/YY orMM/DD/YY.

```
A SF LIN OPE OPERAND          LVTY CONDITION
BB 100                          99IT COND1
BB 110                          OR COND2
BB 120 AD  ID DELCO1 DELCO2
```

AD0: Century positioned from the DAT-CTY field initialized to '19' and that can be modified by the user.

AD1: Century set to '19' if System year is less than the value in DAT-CTYT field (default = '61'), or to 20 otherwise.

AD2: Century set to '20' if System year is less than the value in DAT-CTYT field (default = '61'), or to 19 otherwise.

OPERATORS FOR TIME PROCESSING

TIM: Hour display in 'HHMMSS' format from the EIBTIME field under CICS, from the TIME field with other hardware.

Example:

```
A SF LIN OPE OPERAND          LVTY CONDITION
BB 100 TIM FIELD1
```

TIF: 'HHMMSS' format changed into 'HH:MM:SS'.

Example:

```
A SF LIN OPE OPERAND          LVTY CONDITION
BB 100 TIF FIELD1 FIELD2
```

Inserting and Replacing Automatic Sub-Functions

In a Business Component, procedural code can be added only by insertion in the automatic functions.

EXCEPTIONS:

Function 80: Particular case documented at the end of this section.

Function 81: This function is the only one which follows the standard rules of Structured Code; functions and sub-functions are sorted in alphabetical order.

STRUCTURE OF A GENERATED BUSINESS COMPONENT

A Business Component is structured as follows:

A first part includes the processing of each Logical View.

This processing calls, via PERFORM, elementary processing for each Logical View. The elementary processing is generated in the second part of the Program.

This elementary processing calls in turn, via PERFORM, the processing of each Segment. This processing is generated after the elementary processing on Logical View.

NOTE: To facilitate your reading of the generated code, consult the list of generated labels in the 'Pacbench C/S - Business Logic' manual.

TYPES OF BLOCKS TO BE USED FOR INSERTING PROCEDURAL CODE IN A BUSINESS COMPONENT

You must insert procedural code in special types of blocks: *C, *A, *P, *R, and *B, inside the generated functions.

- You use the *C block to:
 - declare an insertion or a replacement in the Business Component,
 - identify the Logical View for which you are going to insert or replace code.

A *C block must be defined on a level 05 and cannot contain statements. The statements must be written on lower hierarchical levels via the following types of blocks:

- You use the *A block to insert before the automatic processing.
- You use the *P block to insert after the automatic processing.
- You use the *R block to replace the automatic processing.
- You use the *B block to insert in the elementary processing called by PERFORM. This insertion is performed after the automatically generated label or the associated conditioning.

The *B Block concerns the following processing:

- Fvunn-CCNX
- Fvunn-DCNX
- Fvunn-OPEN
- Fvunn-CLOS
- Fvunn-TRER-ffnn
- Fvunn-TRDT-CATx
- Fvunn-TRVW-CATx
- Fvunn-CHKD-CATx
- Fffnn-CHCK-CATx
- Fffnn-UPDT-CATx
- Fffnn-SLCT-CATx

INSERTION RULES IN A BUSINESS COMPONENT

- You cannot create two sub-functions, in -P lines, at the same insertion point (same generated function) with the same type of block.
- The levels which are not imposed (for the *A and *P blocks) follow the standard rules detailed in the 'Structured Code' manual, chapter 'Modifying the Procedure Division'.

Examples:

LEV.	TYPE	CONDITION
08	*A	BEGV : the automatically generated code corresponding to the beginning of the Logical View is included in the level 08

10 *A BEGV : the automatically generated code corresponding to the beginning of the Logical View is not included in this level 10.

- Each insertion/replacement of processing on a Logical View (check/update, selection, transfer...) is limited to 36 sub-functions.. Otherwise the user sub-functions would merge.
- The replacement of a Data Element check or the insertion of such a check is limited to only one sub-function per Data Element and 90 lines of generated code for each of these sub-functions.

REMINDER: These checks are generated if the option CHECKSERV=YES is indicated on a -GO line of the Business Components Dialogue.

This option also conditions the transfer of data from the Logical View to the SQL Segments, depending on the presence of the Data Element.

- When you replace standard code, the automatically generated label no longer appears. To keep this label, code a 'Y' operator on the following line.
- The insertion rules are detailed for each automatic processing in the following pages.

NOTE: 'type' means you can enter one of the following values:

- *A - insertion before the automatically generated code,
- *R - replacement of the automatically generated code,
- *P - insertion after the automatically generated code.

```
BEGIN SERVER
LEV. TYPE CONDITION
05 *C BS type
END SERVER
LEV. TYPE CONDITION
05 *C ES type
DECLARATIONS OF SQL CURSORS
LEV. TYPE CONDITION
05 *C SQ level not imposed for A, P type
SQL ORDERS WHENEVER
LEV. TYPE CONDITION
10 type SQLW level not imposed for A, P type
SQL ORDERS DECLARE CURSOR
LEV. TYPE CONDITION
10 type SQLD ffnn level not imposed for A, P type
ffnn represents the code of the SQL table or view
LOGICAL VIEW (DECLARATION OF PROCESSING)
LEV. TYPE CONDITION
05 *C vunn
vunn is the code of the Logical View
```

This declaration of processing is required when the validation/update, selection on a view as well as the transfer elementary processing, database connection and disconnection are implemented by procedural code.

```
LEV. TYPE CONDITION
05 *C vunn type
```

This special declaration acts on ALL the generated functions of the Logical View; its impact varies according to the value ('type') entered after the code of the Logical View:

*A: insertion before all the functions of the Logical View

*P: insertion after all the functions of the Logical View

*R: replacement of all the functions of the Logical View

```
BEGIN LOGICAL VIEW PROCESSING
LEV. TYPE CONDITION
10 type BEGV          level not imposed for A,P type
VALIDATION/UPDATE OR SELECTION
LEV. TYPE CONDITION
10 type t             level not imposed for A,P type
t is the type of processing ('CHUP' for validation/update
processing, 'SELC' for selection processing, 'USER' for the
procedural code).
VALIDATION/UPDATE OR SELECTION BY CATEGORY
LEV. TYPE CONDITION
15 type t c          level not imposed for A,P type
t is the type of processing as seen above.
c is the category ('CATA' for before the repetitive, 'CATR'
for the repetitive, 'CATZ' for after the repetitive).
VALIDATION/UPDATE OR SELECTION BY CATEGORY/OPERATION
LEV. TYPE CONDITION
20 type t c o        level not imposed for A,P type
t is the type of processing as seen above.
c is the category as seen above.
o is the operation code ('SRVx' or x is the operation
code).
ERROR ON SERVICE REQUEST PROCESSING
LEV. TYPE CONDITION
10 type ERRV          level not imposed for A,P type
END OF LOGICAL VIEW PROCESSING
LEV. TYPE CONDITION
10 type ENDV          level not imposed for A,P type
CONNECTION TO SQL DATABASES VIA THE VIEW
LEV. TYPE CONDITION
10 type CCNX          level not imposed for A,P type
vnnn is the code of the view.
DISCONNECTION FROM SQL DATABASES VIA THE VIEW
LEV. TYPE CONDITION
10 type DCNX          level not imposed for A,P type
DATA TRANSFER
LEV. TYPE CONDITION
10 type ssss          level not imposed for A,P type
ssss is the direction of the transfer:


- TRDT for the transfer of data from the view toward the access Segments
- TRVW for transfer in the opposite direction.

```

TRANSFER OF DATA BY CATEGORY

LEV. TYPE CONDITION

15 type ssss c level not imposed for A,P type
ssss is the same as above.

c is the category code ('CATA' = before repetitive,
'CATR' = repetitive, 'CATZ' = after repetitive).

OPEN VIEW FILES

LEV. TYPE CONDITION

10 type OPEN level not imposed for A,P type

OPEN INDEXED FILES

LEV. TYPE CONDITION

15 type OPEN sd level not imposed for A,P type

sd is the Data Structure code.

CLOSE VIEW FILES

LEV. TYPE CONDITION

10 type CLOS level not imposed for A,P type

CLOSE INDEXED FILES

LEV. TYPE CONDITION

15 type CLOS sd level not imposed for A,P type

sd is the Data Structure code.

VALIDATION, UPDATE, SELECTION ON SEGMENTS

LEV. TYPE CONDITION

10 type tttt ssn level not imposed for A,P type

tttt can have the following values: 'CHCK' for validation
processing, 'UPDT' for update processing, 'SLCT' for
selection processing. ssn is the Segment code.

VALIDATION, UPDATE, SELECTION ON SEGMENTS BY CATEGORY

LEV. TYPE CONDITION

15 type tttt ssn c level not imposed for A,P type

tttt and ssn are the same as above.

c is the category code ('CATA' = before repetitive,
'CATR' = repetitive, 'CATZ' = after repetitive, 'CATT' =
repetitive (access to only one occurrence).

VALIDATION, UPDATE, SELECTION ON SEGMENTS BY CATEGORY
AND BY TYPE OF PROCESSING

LEV. TYPE CONDITION

20 type tttt ssn c t level not imposed for A,P type

tttt, ssn and c are the same as above.

t is the type of processing and can have the following values: 'ALIM' for loading
keys, 'CALL' for physical processing call, 'ERRS' for error processing.

EXAMPLES:

In order to insert processing before the loading of the validation of the DE00
Segment in the heading category, it is necessary to code:

20*A CHCK DE00 CATA ALIM

To replace the service processing of the VU01 Logical View of the repetitive
category, it is necessary to code:

05*C VU01

20*R SELC CATR SRVA

REPLACEMENT OF AUTOMATIC PROCESSING IN F80

To replace the automatic processing in function 80, enter the Segment code in the *R type of block and in the CONDITION FOR EXECUTION 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, refer to the 'SQL Databases' manual, chapter 'SQL Accesses', subchapter 'Customized SQL Accesses'.

REPLACEMENT OF F80-OK AND F80-KO SUB-FUNCTIONS

To replace the F80-OK or F80-KO sub-function, you must code :

- 80 in the function code,
- 99 in the sub-function code,
- 10*R in LEVEL/TYPE.

Inserting a User Service Processing

To add a User Service on a Logical View, you must code:

- '15*C' in the LEVEL/TYPE field,
- 'USER code of the User Service' in the CONDITION FOR EXECUTION field.

Maximum length of the User Service code = 24 characters.

NOTE: With a TUI Client, you must make sure that the code of the User Service corresponds to that set by the TUI Client in the TECH-SRVUSR field of the communication area.

This code must be inserted in a Logical View, i.e. in any sub-function which depends on a function where 05*C has been coded in LEVEL/TYPE and the code of the logical view has been entered in the CONDITION FOR EXECUTION field.

EXAMPLE: LVTY CONDITION
 05*C VL00
 15*C USER UserServiceCode

>>>> The User Service is documented in details in the 'Pacbench C/S - Business Logic' manual.

DL/1 Secondary Index Management

The generation of Segment accesses via secondary indexes is based on the description of Database Blocks in the Specifications 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 on the '-G' line 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 'Comments' (-GC) 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 selection 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 target Segment code must be the same than the source Segment code.

Chapter 4. Folder / Folder View

Definition of a Folder / Folder View

The Definition of a Folder or of a Folder View is documented in details in the 'Pacbench C/S - Business Logic' manual.

```

+-----+
!               ENGLISH DOCUMENTATION LIBRARY           PDSG.VDOC.ACC.439  !
!               1 2                                     !
! FOLDER CODE           FOCLNT                         !
! FOLDER NAME.....: CLIENT FOLDER 3                   !
! FOLDER TYPE.....: F 4           FOLDER               !
!                                                         !
!                                                         !
!                                                         !
! LOCK OPTION.....:N 5                                 !
! PAGING MODE.....:N 6                                 !
!                                                         !
! VARIANTS.....:3 7 1 8  MICROFOCUS OS/2              !
! CONTROL CARD OPTIONS FRONT & BACK..:9 10            !
! EXTERNAL NAME.....:SDCLIENT 11                       !
! ERROR MESSAGE SERVER.....:CLLERR 12                  !
!                                                         !
! EXPLICIT KEYWORDS..: 13                               !
! UPDATED BY.....: F060336  ON : 09/17/2001 AT: 11:07:34  LIB : ACC !
! SESSION NUMBER.....: 0377  LIBRARY.....: ACC  LOCK.....: !
!                                                         !
! O: C1 CH: 0 foclnt           ACTION:                 !
+-----+

```

```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY           PDSG.VDOC.ACC.439  !
!                               1 2                                     !
! LOGICAL VIEW CODE                FVCLNT                             !
!
! LOGICAL VIEW NAME.....: 3 Folder View      (FOVLNT)             !
! FOLDER TYPE.....: 4 FV          FOLDER VIEW                       !
!
!
!
!
!
!
!
!
! CONTROL CARD OPTIONS FRONT & BACK..:9 10                         !
! CLASSES PREFIX.....:FOLDER 11                                     !
! FOLDER.....:FOCLNT 12                                           !
!
! EXPLICIT KEYWORDS...: 13                                         !
! UPDATED BY.....:          ON :          AT:  : :          LIB : ACC !
! SESSION NUMBER.....:0377    LIBRARY.....: ACC  LOCK.....:      !
!
! 0: C1 CH: 0 fvoclnt                ACTION:                       !
+-----+

```

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	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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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.
5	1		LOCK OPTION
		BLANK	Retrieval of the value specified at the higher level.
		N	No lock option (Default value on the Dialogue level).
		O	Lock option, the unlock order is executed before the updates.
			The image initially loaded from the Folder/Folder View is compared with the current image in the Database. If both images are identical, the update is accepted.
		P	Lock option, the unlock order is executed after the updates.
			This value creates an exclusive appropriation of the Folder/Folder View during the update.
6	1		PAGING MODE
		BLANK	The value specified at the Dialogue level is retrieved
		N	Non-extend paging mode. (Default value at the Dialogue level). Allows to page forward or backward through a predefined population. Each paging operation (forward or backward executes a read request and its result replaces that of the previous read.
		E	Extend-mode paging. Allows to page forward through a predefined population and the result of the read request adds to the preceding one.
7	1		TYPE OF COBOL TO GENERATE
		N	No generation
			ON-LINE SYSTEMS DEVELOPMENT FUNCTION:
		X	IBM MVS COBOL II
		O	IBM MVS COBOL/VS

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	X	IBM VSE COBOL II
	1	IBM VSE COBOL/VS
	3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT COBOL
	O	IBM AS/400 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 OR VSE 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).
	3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT 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
8	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
	0	Variants 0, 1, X for CICS (IBM) (programs and BMS maps).
		Variant 2 for IBM 36, monochrome map.
		Variant 3 for PC/MICROFOCUS (MS/DOS).
		Variants 4, 5, 6 for QUESTAR.

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			(program and MFS map for the client).
			Variant 3 for MICROFOCUS OS/2.
			Variant F for TANDEM TUXEDO.
			Variant 8 for UNISYS-A Open/OLTP.
		C	Variants 0, 1 for MULTI-SCREEN CICS or variant X for CLIENT ONLY.
			Variant 4 for MULTI-SCREEN GCOS7.
			Variant 5 for MULTI-SCREEN GCOS8.
			Variant R for TUXEDO (from release 6.2 and TUXEDO (from release 6.2 and higher).
		2	Variants 0, X for CICS (IBM) (program and BMS color map).
			Variant 3 for MICROFOCUS UNIX.
		3	Variants 0, X for IMS (program and MFS color map).
			Variant 3 for MICROFOCUS.
		4	Variant 3 for IBM VISUALAGE COBOL.
9	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 '\$' is used to disable the Business Component generation and to enable the Services Manager generation.
10	1		CONTROL CARDS AFTER PROGRAMS
			Option code which selects the JCL lines to be inserted after each generated program.
		\$	No generation of program.
11	8		EXTERNAL NAME or CLASS PREFIX
			For a Folder or a Communications Monitor, this field must contain its external name.
			For a Folder View, this field contains the class prefix.
			Default value: the occurrence code.
			For a Folder extraction via the GVC command, the value recognized is the occurrence code even if another value was entered in the field.
			In a Folder View, this field contains the class prefix. Value required in this field.
			NOTE: The external name given with the extraction of a Folder View - via the GVC command - is the occurrence code of the Folder View.
12	8		ERROR MESSAGE SERVER or FOLDER
			For a Folder, this field includes the code of the Error Message Server.
			For a Folder View, this field includes the code of the Folder to which it is linked.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			NOTE: 6 character-value to align on the left.
13	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'.

Description of a Folder / Folder View

The Description of a Folder or of a Folder View is documented in details in the 'Pacbench C/S - Business Logic' manual.

NOTE: The description of a Folder or of a Folder View can only be made with the VisualAge Pacbase WorkStation (Pacbench module).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
6	14	SOURCE DATA ELEMENT
		This field must be filled in a line calling a depending or a reference node for each Data Element in the node's key whose code is different from that of the corresponding Data Element in the parent node's key.
		Code of this Data Element in the parent node.
		NOTE: This field must be filled on a line number other than 00.
7	6	NODE'S KEY
		This field must be filled on a line calling a depending or a reference node for each Data Element in the node's key whose code is different from that of the corresponding Data Element in the parent node's key.
		Code of this Data Element in the depending or the reference node.
		NOTE: This field must be filled on a line number other than 00.
8	1	NODE TYPE (REQUIRED IN CREAT)
		(required for line zero).
	R	Root type node.
	D	Dependent type node.
	L	Reference type node.
9	1	TECHNICAL FIELD
		The contents of this field must not be changed.
10	6	BUSINESS COMPONENT CODE
11	4	LOGICAL VIEW CODE
		The Logical View identified in this field must appear in the Segment calls of the Business Component set in the previous field.
12	1	NUMBER OF SUB-SCHEMA
		This field can be used for a Logical View called in a reference node.
	BLANK	All the Data Elements In the Logical View
	1 to 9 0	Sub-schema number 1 to 9 Identifies the sub-schema number 10
		NOTE: The sub-schemas are defined in the Description of the Logical View (CH: S....CE).
13	2	CARDINALITY
		The cardinality indicates the minimum and maximum number of occurrences of the depending or reference node that can be linked to the parent node.
	01 0N	This cardinality is not valid for a reference node.
	11 1N	This cardinality is not valid for a reference node.

Comments (-GC)

The 'Comments' screen (-GC) enables you to write and edit text in relation to one selected instance. This screen is available on all entities.

Each comments line is made up of a 60-character field containing text and a 'Type of Line' field for particular purposes (comment, call of a P.I.A, indication of an alias).

CALLING A P.I.A.

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

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

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

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

Additional lines cannot be inserted between called lines.

For more information see the 'Character Mode User Interface' guide, chapter 'Documentary Facilities', subchapter 'Parameterized Input Aids'.

PREREQUISITES

The instance must be defined prior to being documented.

The called P.I.As must also exist.

LINES COPY

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

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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
4	60	DESCRIPTION - FIRST PART
		If no P.I.A. is called, you can use the whole field.
		On a blank-type line, you enter a free comment.
		In order to select only certain lines of the comments of an instance, use print option -EG and indicate the \$OFF command at the left end of each line to be ignored. To re-activate the -EG option on a line, indicate a \$ON command, left-justified, after the last line to be ignored.
		On an A-type line, enter the COBOL Segment name. (see chapter 'Generation of Copy Book', subchapter 'Description of VA Pac PIA 'Data'', value 'A*' in the 'Data Structure Code in Gener. Descr.' field.)
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.

Folder (Dialogue) Generation Options (-GO)

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

- on an SQL Database Block, you specify options such as the prefixing, the transformation of lowercase into uppercase...
See the 'SQL Databases' Manual.
- on a Segment, you can inhibit the conversion of lowercase letters into uppercase letters on customized SQL accesses (with the UPPER=NO option).
See the 'SQL Databases' Manual.
- on a Program, you can change the function/subfunction number of the date transformation function (F9520 by default), with the DATPRO=ffss option.
See the 'Batch Applications' manual.
- For all entities managed by the Pacbench C/S Function except blank-type or MW-type entities, you can indicate various generation options.
See the 'Business Logic' Volume of the Developer Documentation dedicated to eBusiness applications.

All these Manuals can be consulted at our internet address:

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

This link to the VA Pac Documentation web site requires a specific login with password. If you are not a registered user, send your login request to vapacsup@fr.ibm.com.

CALLING A P.I.A.

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

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

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

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

Additional lines cannot be inserted between called lines.

For more information see the 'Character Mode User Interface' guide, chapter 'Documentary Facilities', subchapter 'Parameterized Input Aids'.

PREREQUISITES

The instance must be defined prior to being documented.

The called P.I.As must also exist.

LINES COPY

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

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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
4	60	DESCRIPTION
		If no P.I.A. is called, you can use the whole field.
		On an O-type line, Enter the option followed by an equal sign and the value (with no space).
5	40	DESCRIPTION / SECOND PART
		This field is specific to a P.I.A. call.
		With value 'C2' in the OPERATION CODE field, the cursor automatically tabs to the first position of this field.
		This field is initialized with 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.

Chapter 5. TUI Client

Introduction to TUI Applications

The purpose of Pacbench C/S is to develop and generate automatically TUI applications.

This functionality ensures :

- the description of all Client components using the Screen/CS entities of the VisualAge Pacbase Metamodel.

The occurrences thus described are managed by the VisualAge Pacbase Specifications Dictionary.

- the automatic generation of TUI screen layouts.
- the generation of Client components as well as the TUI screen flow and the Client/Server components flow.

the Client Program allows to submit service requests to the Server program and manages the end user's interface.

The Client Program processes :

- the reception of input data,
- the validation of this data,
- the management of detected errors and the display of error messages,
- the data access or calculation service calls,
- the formatting and display of the application data.

The Client program consists of three main phases : the reception of the input physical message from the TP Monitor (contents of the screen), the call of services and the display of output data.

The client Program is described via an occurrence of the SCREEN C/S entity and generated by Pacbench C/S; it is the TUI Client.

GENERAL PRINCIPLES

The standards and elements common to all Client components are defined at the Dialogue level; therefore, the presentation and operation of the screens will be homogenous.

For example, the following elements are defined at the Dialogue level :

- TUI Screen size (number of lines, of columns),
- Display attributes (intensity, presentation, color),
- Communication area.

These general characteristics are default options of the Dialogue components.

A TUI Screen is made up of two types of elements that describe the screen :

- Calls of Data Elements defined in the Specifications Dictionary.
- Labels specific to the Screen.

- Each field has :
 - A position in the Screen (relative or absolute),
 - A name built directly from the Specifications Dictionary (for Data Elements),
 - Display characteristics (intensity, presentation, color).

The Data Elements display options are set at the Screen definition level in order to be homogenous. They can be changed at the level of each Data Element call in the Screen.

These elements make up the TUI Screen layout.

The Client component (local data) can ensure the processing of data external to the Screen or this can be done by the call of one or more Business Component(s).

This processing is determined by the call of Segments which describe data required in the Client component by indicating for each of these its physical characteristics (organization ...) and use (read, write...).

Complementary procedural code can be added with Structured Language.

GENERATION

Once this data has been defined, the system ensures :

- An automatic generation of each TUI Client component.

This generation is adapted to the system and TP Monitor characteristics according to the variant specified at the Library, Dialogue or C/S Screen level.
- The automatic generation of the processing for each Client component (one Program per C/S Screen).

This Program is generated in COBOL and is adapted to the hardware and TP monitor in use according to option defined at the Library, Dialogue or Screen level.

Definition of a TUI Client (Dialogue)

The purpose of this screen is to define either a Dialogue or a C/S Screen. This screen is accessed by entering the following values in the CHOICE field :

CH : Odd (Dialogue) or CH : Oddeeee.

This screen contains a field in which you indicate the component type ('C' for Client, 'MC' for Client Monitor, etc.).

For a C/S Screen, its Dialogue includes the presentation characteristics and the default values of the attributes of the possible fields.

For a Dialogue, all fields are input fields, the values displayed by default correspond to the general characteristics.

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

The physical characteristics of the Screen (size, presentation, attributes, etc.) do not apply to the Business Component. The Business component includes file or table accesses and specific code, whereas the -CE, -L and -M screens are entered for specifying TUI client screens.

>>>> Mapping operations are fully documented in the 'WorkStation User Interface' guide WorkStation Reference Manual, Chapter 'Lo, chapter 'Simulation Window'.

If you are not using the VisualAge Pacbase WorkStation, then refer to the 'On-Line Systems Development' manual, chapter 'Definition and Description of a Dialogue or Screen'.

PREREQUISITE DESCRIPTIONS

To define a C/S screen, it is required that the Dialogue, corresponding to the first two characters of its code, should be defined.

ACTION CODES

The possible different value of the action code are listed in the 'Character Mode User Interface' guide.

NOTE ABOUT DELETION

Deletion is possible only if there are no definition or description lines (Dialogue complement, Screen, Documentation, areas, Segments, etc.).

```

+-----+
!           ENGLISH DOCUMENTATION LIBRARY           *PDMC.VDOC.ACC.3!
!                   1 2
! ON-LINE DIALOGUE CODE                   ZR
!
! DIALOGUE NAME.....: 3 FLIGHT RESERVATION (CLIENT)
!
! SCREEN TYPE.....: 4 C           5 CLIENT COMPONENT
! SCREEN SIZE (LINES, COLUMNS) .....: 6 24       7 080
! LABEL TYPE, TABS, INITIALIZATION...: 7 L           9 01       10
! HELP CHARACTER SCREEN, DATA ELEMENT:11 10       12 11
!
!
!                   LABELS  DISPLAY  INPUT  ER.MESS.  ER.FL!
! INTENSITY ATTRIBUTE .....:13 N       14 N       15 N       16 N       17 N !
! PRESENTATION ATTRIBUTE .....:18 N       19 N       20 N       21 N       22 N !
! COLOR ATTRIBUTE .....:23 W       24 W       25 W       26 W       27 W !
!
! TYPE OF COBOL AND MAP TO GENERATE...:28 X 29 0   IBM VS2 CICS (PRO. MAP BMS) !
! CONTROL CARD OPTIONS FRONT & BACK...:          30-31 (PROGRAM) 32       33       (MAP)!
! EXTERNAL NAMES .....:          34       (PROGRAM) 35       (MAP)!
! TRANSACTION CODE.....:36
!
! EXPLICIT KEYWORDS...:37
! UPDATED BY.....:          ON :          AT: : :          LIB : ACC
! SESSION NUMBER.....: 0378          LIBRARY.....: ACC       LOCK.....:
!
! O: C1 CH: 0 zr          ACTION:
+-----+

```

```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY                               *PDMC.VDOC.ACC.3!
!                               1 2
! ON-LINE DIALOGUE CODE                               ZR77PI
!
! DIALOGUE NAME.....: 3 PILOT INITIALIZATION
!
! SCREEN TYPE.....: 4 C          CLIENT COMPONENT
! SCREEN SIZE (LINES, COLUMNS) .....: 5 24      6 080
! LABEL TYPE, TABS, INITIALIZATION...: 7 L        8 01    9
! HELP CHARACTER SCREEN, DATA ELEMENT:10 10     11 11
!
!                               LABELS  DISPLAY  INPUT  ER.MESS.  ER.FL!
! INTENSITY ATTRIBUTE .....:12 N      13 N      14 N      15 N      16 N !
! PRESENTATION ATTRIBUTE .....:17 N     18 N      19 N      20 N      21 N !
! COLOR ATTRIBUTE .....:22 W          23 W      24 W      25 W      26 W !
!
! TYPE OF COBOL AND MAP TO GENERATE...:27 X 28 0   IBM VS2 CICS (PRO. MAP BMS) !
! CONTROL CARD OPTIONS FRONT & BACK...:29 30      (PROGRAM) 31 32 (MAP)!
! EXTERNAL NAMES .....:33 WSPPIP (PROGRAM) 34 WSPPIP (MAP)!
! TRANSACTION CODE.....:* WSPI 35
!
! EXPLICIT KEYWORDS...:36
! UPDATED BY.....:          ON :          AT: : :          LIB : FCI
! SESSION NUMBER.....: 0388      LIBRARY.....: FCI    LOCK.....:
!
! 0: C1 CH: 0 zr77pi          ACTION:
+-----+

```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			COMPLETE CODE
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	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

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

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
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 into 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
			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

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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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:
		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.

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

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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.
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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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
		O	IBM MVS COBOL/VS
		X	IBM VSE COBOL II
		1	IBM VSE COBOL/VS
		3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT COBOL
		O	IBM AS/400 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 OR VSE 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).
		3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT 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

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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	S	Variant F for 'SERVER' type program. or variant R for 'SERVICE' type program.
		PACBENCH CLIENT/SERVER
		TP MONITOR
	0	Variants 0, X for CICS (IBM) (Program and BMS map for the client).
		Variant 3 for MICROFOCUS MS/DOS.
		Variant F for TANDEM Pathway.
		Variant R for TUXEDO.
		Variant 7 for VPLUS.
		Variant U for UNISYS-2200.
		Variant 8 for UNISYS-A.
	1	Variants 0, X for IMS.
		(program and MFS map for the client).
		Variant 3 for MICROFOCUS OS/2.
		Variant F for TANDEM TUXEDO.
		Variant 8 for UNISYS-A Open/OLTP.
	C	Variants 0, 1 for MULTI-SCREEN CICS or variant X for CLIENT ONLY.
		Variant 4 for MULTI-SCREEN GCOS7.
		Variant 5 for MULTI-SCREEN GCOS8.
		Variant R for TUXEDO (from release 6.2 and TUXEDO (from release 6.2 and higher).
	2	Variants 0, X for CICS (IBM) (program and BMS color map).
		Variant 3 for MICROFOCUS UNIX.
	3	Variants 0, X for IMS (program and MFS color map).
		Variant 3 for MICROFOCUS.
	4	Variant 3 for IBM VISUALAGE COBOL.
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 '\$' 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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Option code that identifies the job card to be inserted before each generated Screen or C/S Screen map.
	\$	No generation of map.
		NOTE: This field is not used in a Pacbench C/S development with specification of Folder.
		Business Component / single-view (with no specification of a Folder).
		Option code which selects the JCL lines to be inserted before the generated Services Manager. The value '\$' is used to disable the generation of the Services Manager and to enable the Business Component to be generated.
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.
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).
		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 and Architectures' manual.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
35	8		TRANSACTION CODE
			ON-LINE SYSTEMS DEVELOPMENT:
			CICS: 4-character transaction code (default).
			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.
			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'.

Call of Data 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.

```

+-----+
!                ENGLISH DOCUMENTATION LIBRARY                *PDMC.VDOC.ACC.3!
!                1
! SCREEN CALL OF ELEM... ZR77PI PILOT INITIALIZATION
! 2 3   4       5 6 7  8 9  17      19  21  23      24 25
!                16  18      20  22
! A LIN : D.ELEM . PHYSICAL ATTRIBUTES . VALIDATION UPDATE . DISPLAY
!       :       . P LN COL N L C HR VR . P V U UPD TARGET . S SOURCE   LV!
!-----+
! . 020 : ZR0000 . A 01 001 S . . . . .
! . 050 : ZR77PI . A 03 030 T . . . . .
! . 100 : REPET . A 07 001 R 1 10 . . . . .
! . 110 : COACT . . 003 V . . I V A . . a
! . 111 : . . . . . V M . . M
! . 112 : . . . . . V C . . C
! . 113 : . . . . . V M . . m
! . 114 : . . . . . V C . . c
! . 120 : NOPIL . . 002 V . . R T VU03 . . VU03
! . 130 : NOMPIL . . 002 V . . R X VU03 . . VU03
! . 170 : DAEMB . . 002 V . . R X VU03 . . VU03
! . 180 : COQUA . . 002 V . . R X VU03 . . VU03
! . 190 : QUSAM . . 002 V . . R X VU03 . . VU03
! . 192 : DAQUA . . 002 V . . R X VU03 . . VU03
! . 194 : DASAL . . 002 V . . R X VU03 . . VU03
! . 200 : BASEC . A 19 001 Z . . . . .
!
! 0: C1 CH: 0 zr77pi CE
+-----+

```



```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY                               *PDMC.VDOC.ACC.3!
!                               1
! SCREEN CALL OF ELEM... ZR77PI PILOT INITIALIZATION
! 2 3   4       5 6 7   8 9 17   10 12 14   28
!                               18 11 13 15   29
! A LIN : D.ELEM . PHYSICAL ATTRIBUTES . LABEL
!       :       . P LN COL N L HR VR IN PR CO . T LITERALS
!
! . 020 : ZR0000 . A 01 001 S .
! . 050 : ZR77PI . A 03 030 T .
! . 100 : REPET . A 07 001 R 1 10 .
! . 110 : COACT . 003 V .
! . 111 : . . .
! . 112 : . . .
! . 113 : . . .
! . 114 : . . .
! . 120 : NOPIL . 002 V .
! . 130 : NOMPIL . 002 V .
! . 170 : DAEMB . 002 V .
! . 180 : COQUA . 002 V .
! . 190 : QUSAM . 002 V .
! . 192 : DAQUA . 002 V .
! . 194 : DASAL . 002 V .
! . 200 : BASEC . A 19 001 Z .
!
! 0: C2 CH: 0 zr77pi CE
+-----+

```


NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			It uses by default the format defined at the Dictionary level. If it is absent:
			. 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.
			DATA ELEMENT GROUP DEFINING A CATEGORY:
			Its definition in the Dictionary is optional.
			SPECIAL DATA ELEMENTS:
		ERMSG	This allows you to allocate the placement of error messages. By default, its length is 72 characters and it can be modified if the ERMSG Data Element is defined in the Dictionary with a different format.
			It cannot be repeated vertically.
			Its required nature in the Screen is P.
		PFKEY	Create a function key to:
			. orient the Dialogue to another Screen. The Screen to which the branching is done and the numbers of the function keys are indicated in the UPDATE and SOURCE fields.
			. initialize the operation code of the Screen (OPER). The initialization character of the OPER variable and the number of the function key are indicated in the UPDATE and SOURCE fields.
			Its position has no influence on the composition of the Screen and can be anything.
			The value of the function key depends on the generation variant. Generally, it must be numeric and correspond to the number of keys available on the keyboard.
			CICS: It is also possible to use:
			A1 for the PA1 key,
			A2 for the PA2 key,
			EN for the ENTER key,
			00 for the CLEAR key.
			TUXEDO, IMS: It is also possible to use:
			EN for the ENTER key.
			HP3000: Values ranging from 1 to 8 may be used for the PFkeys.
		*PASWD	In the On-Line Systems Development function, allows, under IMS, to set a password at the Screen level (generation of the PASSWORD macro at the MID level). This Data Element must be described in the Dictionary and have a length less then or equal to 8 characters.
			CALL OF ANOTHER SCREEN:
			Code of the called Screen (with the 'S' value in the DATA ELEMENT NATURE field).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		If the called Screen contains a Screen call, this latter call is ignored in the call of the first level.
		SCREEN TITLE:
		Screen code: by indicating the T value in the DATA ELEMENT NATURE field, the name of the Screen will appear. (This operation can be used in order to describe a menu-type Screen).
		FIXED LABEL:
		In the Data Element list of a Screen (-CE), a fixed label (Data Element Nature in the 'L' screen) can be created.
		Leave the Data Element code field blank.
		Indicate the label in the LITERALS field which is available with the C2 operation code.
		Its characteristics, other than its value, can be modified on the Screen obtained by the C1 operation code.
		All its characteristics can be changed on the Screen Layout -L or Mapping -M screens.
		The '/' character is used as a delimiter at the end of the label. However, it is possible to use '/' inside the label. The last '/' is considered a delimiter.
		A fixed label is limited to 30 characters, except if it is made up of one repeated character. In this case, the limit is 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.

NUM	LEN	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
			The possible natures are grouped as follows:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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 the two previous cases, if the Data Element does not have a conventional 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	Label of the Data Element only.
		L	Literal. (See DATA ELEMENT OR SCREEN CODE field)
		T	Title of the Screen of which the name is indicated in the DATA ELEMENT OR SCREEN CODE field. (See DATA ELEMENT OR SCREEN CODE field)
			GROUP DATA ELEMENTS DEFINING A CATEGORY
			The Data Elements of the Screen fall into three categories: top, repetitive, bottom.
			All Data Elements appearing after the group Data Element belong to the category defined for that group until the next group Data Element or the end of the list. The Data Elements appearing before the first group Data Element belong to the top category.
		R	Repetitive.
		Z	Bottom.
			There can be a maximum of one group Data Element 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 category. For the repetitive category, a vertical repetition and a horizontal repetition determine the number of lines of an occurrence.
			SCREEN CALL
		S	Call of another screen. The code of the called Screen is indicated in the place of the Data Element code. It will appear in the cross-references.
9	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),

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		.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.
	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

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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	G	Green.
	T	Turquoise.
	B	Blue.
16	1	CURSOR DEFAULT POSITION
		This field allows you to identify the Data Element on which the cursor is placed by default.
	Y	Positioning of cursor on the Data Element.
		In the absence of this value, the cursor positions on the first keyable Data Element 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.
	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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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
			With this option you can indicate:
			. the use of the Data Element as 'Operation code' for the Screen,
			. or the use of the Data Element as 'Action code' for for the category,
			. or the validation to perform on the Data Element.
			AT THE SCREEN LEVEL:
			The system generates the automatic processing of 6 standard operations - display, update, following screen, exit conversation, same screen, call another screen - in function of the values of an internal operation code (OPER).
		O	Operation code
			This Data Element is identified as the external operation code.
			The correspondence between its possible values and the internal values of OPER is established:
			. either at the Dictionary level (Codification of the Data Element, type-O lines),
			. or at the Screen level for this Data Element (-CE) on a line with "update option" = "V": the OPER value is indicated in the UPDATE TARGET/FIRST PART field, the value of the Data Element in the SOURCE FOR DISPLAY/FIRST PART field. The Data Element must be variable (nature 'V').
			AT THE CATEGORY LEVEL
			The C/S OLSD Function offers the possibility to define four types of actions: creation, modification, deletion, implicit creation or modification, and automatically generate processing in function of the values of an internal action code.
		I	Transaction code
			The Data Element is identified as the external action code. The correspondence between its possible values and the internal values of CATM is established
			. either at the Dictionary level (Data Element Descriptions, I-type lines),
			. either at the Screen level for this Data Element, on a line with the "update option" = 'V': the CATM value is indicated in the UPDATE TARGET/FIRST PART field, the value of the Data Element in the SOURCE FOR DISPLAY/FIRST field. The Data Element must be variable (nature 'V').
			AT THE DATA ELEMENT LEVEL

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Indicate the type of validation for the Data Elements of the Screen:
	T	Validation of the Data Element during modification, creation, deletion.
	X	Validation of the Data Element during creation and modification.
	A	Validation of the Data Element, except during updating.
	BLANK	Update and validation of the Data Element systematically.
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). 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:
	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.
	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 'T' 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).
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 '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'.
23	6	UPDATE TARGET / LAST PART
		(Default Option: Data Element code)
		The default option also works for a modification.
24	1	TYPE OF SOURCE FOR DISPLAY
		1. Indicate the contents of the source field that is used for loading a Data Element's display:
		BLANC
		The source field is a standard VA Pac field
		*
		The source field is not a standard field. In this case, you must use any given name in the source field that will be retrieved as it is in the generated Program.
		2. Used with the update option 'G':
		D
		Ensures the branching to the desired screen after all processing related to the reception of the message (the OPER variable is 0).
		M
		The same functionalities as the 'D' above, but the OPER variable is positioned at the M value.
25	4	SOURCE FOR DISPLAY / FIRST PART

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		In the generated Program associated to the Screen, this field corresponds to the Segment code to which the Data Element updated from the screen Data Element belongs.
		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 this field corresponds to the Segment code to which contains the value of the field (or the PA key) used belongs. to call the requested screen.
		If UPDATE OPTION value = 'V' and with value 'O' or 'T' in the VALIDATION CONDITIONS/SET VARIABLES, field, this field contains the value for the initiali- zation 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	LEVELS OF GENERATED SUB-FUNCTIONS
		PURE NUMERIC FIELD
	06 to 98	Pacbench C/S automatically generates in the Business Component associated with the Client component the processing that depends on the use (in reception and display) of Segments used in this Screen.
		See the 'Structured Code' manual for definitions of function, sub-function and level.
		The hierarchical level is 10 by default.
28	1	TYPE OF LITERAL
		This field defines the contents of the next field (O: C2).
	BLANK	Label displayed on the Screen.
	I	Initial value.
	P	Presentation value used only for simulation.
	A	This option is only used for labels. It allows the initialization of a field of more than 30 characters by the repetition of one character. The structure of the label field is: nnnX where nnn is the number of desired repetitions and X the initialization character.
29	30	DISPLAYED LITERAL
		This field contains either a literal or an initial presentation value. If no TYPE OF LITERAL value is entered in the preceding field, this field contains a fixed value. (See DATA ELEMENT OR SCREEN CODE TO CALL field.)

NUM	LEN	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'.

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:

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

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.

The simulation takes into account the presentation values that were indicated either during the call or those entered on the description of the Data Elements, on P-type lines (see the 'Data Dictionary' manual).

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 a Data Element defined in absolute position: 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),...

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 Call of Elements screen is executed by pressing the corresponding PFkey.

NOTES

The CLEAR key is synonymous with the PFKEY value '00' (except with 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 indicated by a '>' character.

The PFKEY 'nn' number 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, ENTER 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.


```

+-----+
! THE ENGLISH AIRLINE COMPANY          FLIGHTS AND RESERVATIONS          01/01/98!
!
!                                     LIST OF PILOTS
!
!  A  Num  NAME          HIRING  C  SALARY          DATE  QUALIF
!  c  ----  LACOSTES      -----  -  -----        -----
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!  .  .....  .....
!
!
! CHOICE: PI
! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
+-----+

```

```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY                               *PDMC.VDOC.ACC.3!
! ON-LINE SCREEN ADDRESS OF ELEMENTS      ZR77PI  PILOT INITIALIZATION                      !
!
!   LN COL  LTH LIN T D.ELEM R  HR VR S   LN COL  LTH LIN T D.ELEM R  HR VR S   !
!   1  2   28  20 L          7  49   6 192 R DAQUA  1          !
!   1 39   30  22 L ZR0000  7  57   6 194 R DASAL  1          !
!   1 71    8  40 R DAJOU   8   3   1 110 R COACT  2          !
!   3 30   30  50 L ZR77PI  8   6   5 120 R NOPIL  2          !
!   6  3    1 110 L COACT   1  8  13  12 130 R NOMPIL  2          !
!   6  6    5 120 L NOPIL   1  8  27   6 170 R DAEMB  2          !
!   6 13   12 130 L NOMPIL  1  8  35   1 180 R COQUA  2          !
!   6 27    6 170 L DAEMB   1  8  38   9 190 R QUSAM  2          !
!   6 35    1 180 L COQUA   1  8  49   6 192 R DAQUA  2          !
!   6 38    9 190 L QUSAM   1  8  57   6 194 R DASAL  2          !
!   6 49    6 192 L DAQUA   1  9   3   1 110 R COACT  3          !
!   6 57    6 194 L DASAL   1  9   6   5 120 R NOPIL  3          !
!   7  3    1 110 R COACT   1  9  13  12 130 R NOMPIL  3          !
!   7  6    5 120 R NOPIL   1  9  27   6 170 R DAEMB  3          !
!   7 13   12 130 R NOMPIL  1  9  35   1 180 R COQUA  3          !
!   7 27    6 170 R DAEMB   1  9  38   9 190 R QUSAM  3          !
!   7 35    1 180 R COQUA   1  9  49   6 192 R DAQUA  3          !
!   7 38    9 190 R QUSAM   1  9  57   6 194 R DASAL  3          !
!
! 0: C1 CH: 0 zr77pi ADR
+-----+

```

```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY                               *PDMC.VDOC.ACC.3!
! ON-LINE SCREEN ADDRESS OF ELEMENTS      ZR77PI  PILOT INITIALIZATION                      !
!
!   LN COL  LTH LIN T D.ELEM R  HR VR S   LN COL  LTH LIN T D.ELEM R  HR VR S   !
!   10  3   1  110 R COACT  4           12  13   12 130 R NOMPIL  6           !
!   10  6   5  120 R NOPIL  4           12  27   6  170 R DAEMB  6           !
!   10  13  12 130 R NOMPIL  4           12  35   1  180 R COQUA  6           !
!   10  27   6  170 R DAEMB  4           12  38   9  190 R QUSAM  6           !
!   10  35   1  180 R COQUA  4           12  49   6  192 R DAQUA  6           !
!   10  38   9  190 R QUSAM  4           12  57   6  194 R DASAL  6           !
!   10  49   6  192 R DAQUA  4           13   3   1  110 R COACT  7           !
!   10  57   6  194 R DASAL  4           13   6   5  120 R NOPIL  7           !
!   11   3   1  110 R COACT  5           13  13   12 130 R NOMPIL  7           !
!   11   6   5  120 R NOPIL  5           13  27   6  170 R DAEMB  7           !
!   11  13   12 130 R NOMPIL  5           13  35   1  180 R COQUA  7           !
!   11  27   6  170 R DAEMB  5           13  38   9  190 R QUSAM  7           !
!   11  35   1  180 R COQUA  5           13  49   6  192 R DAQUA  7           !
!   11  38   9  190 R QUSAM  5           13  57   6  194 R DASAL  7           !
!   11  49   6  192 R DAQUA  5           14   3   1  110 R COACT  8           !
!   11  57   6  194 R DASAL  5           14   6   5  120 R NOPIL  8           !
!   12   3   1  110 R COACT  6           14  13   12 130 R NOMPIL  8           !
!   12   6   5  120 R NOPIL  6           14  27   6  170 R DAEMB  8           !
!
! 0: C1 CH: 0 zr77pi ADR
+-----+

```

```

+-----+
!                               ENGLISH DOCUMENTATION LIBRARY                               *PDMC.VDOC.ACC.3!
! ON-LINE SCREEN ADDRESS OF ELEMENTS      ZR77PI  PILOT INITIALIZATION      !
!
!   LN COL  LTH LIN T D.ELEM R  HR VR S   LN COL  LTH LIN T D.ELEM R  HR VR S   !
!   14 35   1 180 R COQUA   8           16 49   6 192 R DAQUA  10           !
!   14 38   9 190 R QUSAM   8           16 57   6 194 R DASAL  10           !
!   14 49   6 192 R DAQUA   8           22  4   7 220 L COCHX           !
!   14 57   6 194 R DASAL   8           22 12   2 220 R COCHX           !
!   15  3   1 110 R COACT   9           23  3   72  80 R LIERR           1   !
!   15  6   5 120 R NOPIL   9           24  3   72  80 R LIERR           2   !
!   15 13   12 130 R NOMPIL  9                                     !
!   15 27   6 170 R DAEMB   9                                     !
!   15 35   1 180 R COQUA   9                                     !
!   15 38   9 190 R QUSAM   9                                     !
!   15 49   6 192 R DAQUA   9                                     !
!   15 57   6 194 R DASAL   9                                     !
!   16  3   1 110 R COACT  10                                     !
!   16  6   5 120 R NOPIL  10                                     !
!   16 13   12 130 R NOMPIL 10                                     !
!   16 27   6 170 R DAEMB  10                                     !
!   16 35   1 180 R COQUA  10                                     !
!   16 38   9 190 R QUSAM  10                                     !
! *** END ***
! 0: C1 CH: 0 zr77pi ADR14
+-----+

```

Call of Segments in the TUI Client Component (-CS)

This Screen (-CS) allows you to indicate the type of Segment access (Logical Views and/or Segments describing local data): consultation (validation), update or display associated to the client component.

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

All the Segments which are common to a Data Structure (i.e. have the same Data Structure code) must have the same organization, the same description type and, for a V-type organization, the same external name.

GENERAL CHARACTERISTICS

For each Segment, indicate:

- The category (top, repetitive, bottom).
- The type of logical access (check or update) to perform on the Segment during reception and/or display, and the key loading mode.
- The use in the generated code.
- If the access is conditioned by an access to another Segment.
- The physical organization.

The list of Segment code used in the description of the Screen (List of Screen fields) is displayed in the top category.

NOTES

The Segment organization, along with the variant of the TP monitor indicated in the Screen definition, determines the generation of the Segment description.

For a Segment whose access is done by the procedural code (W-type organization), the description will be generated in the Working Storage Section.

GENERATION OF CLIENT ACCESS

The logical accesses to a Segment are generated independently of the variant, depending on the usage (reception and/or display) of the Segment.

The call of a Segment on the -CS screen causes, at the generation time, a certain number of functions to be introduced in the output source. The functions can be grouped into the following main areas:

- Segment description containing:
 - Input-output fields,
 - DATA DIVISION.
- Logical access code:
 - Reception read file (F25),
 - Write file (F35),
 - Display file access (F60).
- Physical access (F80)

These 3 groups of functions are generated or not according the value of these three fields on the -CS screen:

- Generation limit
- Organization
- Use

The W organization does not generate the F80 function.

The U generation does not generate the F25, F35, and F60 functions.

The P generation does not generate the Segment description.

PREREQUISITE DESCRIPTIONS

The Screen and Segments must be defined.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			For more details, refer to chapter 'Business Component', subchapter 'Call of Segments', section 'Particular case : Error Handling'.
			For the Logical View code being used for the call of a Business Component (X-type organization), the code must be identical to the code used in the associated Business Component.
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.	LINE NUMBER
			PURE NUMERIC FIELD
		00	Only one '00' line number per Segment is permitted. (Default)
			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. Organization different than X.
		U	User-defined Segment access in structured code.
		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.
			For organizations different from X:
		N BLANK	Not used. No access statement will be generated for this Segment in the validation and update processing (default).
		L	Read.
			A simple read command is generated for the Segment.
		E	Existence validation to be executed.
			A read command with the setting of an error code is generated (if it is not found).
		M	Updated Segment (modification only).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		S	Updated Segment (deletion only).
		C	Updated Segment (creation only).
		X	Updated Segment (creation and modification only).
		T	Updated Segment (creation, modification and deletion).
			In the last five 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.
			For organization = X:
		N BLANK	No server call is generated.
		L	Server call for selection.
		E	Server call for validation.
		M	Server call for validation and update.
		T X	Server call for validation, update and selection. NOTE: If the Logical View is called in the repetitive category, it is not possible to display the continuation of the Screen (this means that it is not possible to use the 'S' operator code automatically). Call of a Business Component for validation and selection.
		U	Call of a User Service.
8	1		USE IN DISPLAY
			Organizations different from X:
		N BLANK	Not used.
		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.
			Organization = X:
		N BLANK	No server call is generated.
		A	Call of a Business Component for selection (not compatible with the 'L' value in reception).
		U	Call of a User Service.
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

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		This field can be used as a working area, a literal or Screen field. In the latter, the first 4 characters of the field must correspond to the code of the screen in the Dialogue.
		For example:
		0001-delco for a Screen field,
		'02 ' for a literal,
		ww10-delco for a working area.
11	6	ACCESS KEY (DATA ELEMENT CODE)
		This is the DATA ELEMENT CODE used as the access key to a Segment. it must be entered on the Segment's '00' line. The Data Element entered 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 loading of the associated key (G-ddss-NUSSY).
		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.
		ERROR MESSAGE FILE OR TABLE
		If the error message Segment is a SQL/DS Segment or if it has a V or W organization, the Data Element key must be ERKEY.
12	1	CONTROL BREAK INDICATOR FOR DISPLAY
		For organizations different from X:
		Indicates that a Data Element, which belongs to a key in the repetitive category of a Screen, must remain constant during display.
	BLANK	No control break.
	C	Display break.
		SQL ORGANIZATIONS:
	E	Display control break; causes the generation of an equal condition in the WHERE clause of the SQL 'DECLARE CURSOR' statement (and not >=, which is generated otherwise). Allows optimization of the use of CURSORs in the repetitive category.
	R	Large reading from the root (graphic applications in Folder mode only).
13	1	ORGANIZATION (REQUIRED IN CREAT)

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Required for the '00' LINE NUMBER of the Segment.
	V	Segment Organization: Indexed.
	G	'TABLE'-type Segment. In this case, the generated access corresponds to the call of the Pactables standard access module.
	H	Relational SQL database Segments accessed by SQL :
		The Segment description and the physical accesses are generated; the cursor declaration is generated for the Segments of the repetitive category. The type of relational Database Block is required in the Definition of the Block whose code must be entered in the EXTERNAL NAME field.
		NOTE: Relational organization must be grouped together in the -CS screen.
	W	Segment access is managed by procedural code.
		Only one description in WORKING is generated.
	X	Server call for validation and selection.
		NOTES ON THE ERROR MESSAGE FILE:
		V Organization: Enter the file external name; the VisualAge Pacbase code is not validated.
		W Organization: Working; no access generated.
		H Organization: Indicate the file external name.
		The external name corresponds to the code of the Database Block which contains the relational table indicated in the SEGMENT CODE IN THE DATABASE field.
		X Organization: Call of the server which accesses the error message file.
		The server VisualAge Pacbase code is indicated in the external name.
14	1	DESCRIPTION TYPE
		Allows you to obtain from the Segment description in the database, different descriptions in the generated Program.
	blank	Complete Segment (common and specific parts).
	1	Specific part only. (Default value for a table, ORGANIZATION 'G'.)
	E	TABLE-type Segments: The input format of the Segment's Data Elements is taken into account.
	2	Complete Segment with variable length for indexed organization ('V').
		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.)
		ORGANIZATION V (indexed):
		IBM CICS-VSAM: DDNAME (1 to 8 positions).
		DPS8-TP8: External name (four characters) (generated in the SELECT clause).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Blocking factor (last four characters) (generated in the FD clause).
		Other variant: Logical name, (generated in the SELECT clause).
		ORGANIZATION H (SQL):
		SQL Database Block code, in which is called the table or the view. It is possible to select a table or a view by indicating T or V after the code of the chosen Block. Important: the Block and the type of object must be separated by a blank.
		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).
		ERROR MESSAGE FILE :
		The external name to enter is the code of the Database Block which contains the relational table indicated in the SEGMENT CODE IN THE DATABASE field.
		The description and accesses are generated with the relational table code entered in this field.
		ORGANIZATION G (Tables defined to Pactables):
		In CICS: 4 characters for the DDNAMEs of the files managed by Pactables (these are the characters of the transaction code used to access Pactables files).
		ORGANIZATION X (server call):
		VisualAge Pacbase code of the Business Component.
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:
		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.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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	LEVELS OF GENERATED SUB-FUNCTIONS
		PURE NUMERIC FIELD
	06 to 98	Pacbench C/S automatically generates in the Business Component associated with the Client component the processing that depends on the use (in reception and display) of Segments used in this Screen.
		See the 'Structured Code' manual for definitions of function, sub-function and level.
		The hierarchical level is 10 by default.

TUI Client Dialogue Complement (-O)

For a Client component, this screen is used to indicate additional information required for the generation of programs.

The selected shared communication area can be supplied, whichever TP monitor is used.

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

Several values can be entered in the OPTIONS field, each separated by a blank. During the screen generation, a WARNING type message signals erroneous parameters.

PREREQUISITE

The Dialogue must be defined.

```

+-----+
!           ENGLISH DOCUMENTATION LIBRARY           *PDMC.VDOC.ACC.3!
! DIALOGUE COMPLEMENT....: 1 ZR FLIGHT RESERVATION (client) !
! ! ! ! !
! COMMON AREA-DATA STRUCTURE CODE.....: 2 VR !
! ! ! ! !
! ERROR MESSAGE FILE CHARACTERISTICS !
!           ORGANIZATION....: 3 !
!           EXTERNAL NAME....: 4 !
! ! ! ! !
! FIRST SCREEN CODE OF THE DIALOGUE.....: 5 00ME !
! ! ! ! !
! COMPLEMENTARY COMMON AREA LENGTH.....: 6 !
! ! ! ! !
! CODE OF PSB OR SUB-SCHEMA.....: 7 !
! ! ! ! !
! OPTIONS : 8 MONIT !
! ! ! ! !
! SESSION NUMBER      : 0391  LIBRARY      : ACI !
! ! ! ! !
! 0: C1 CH: 0 zr 0           ACTION: !
+-----+

```

NUM	LEN	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
			NOTE : for a TUI (or character-based) application, it is recommended to use the -CS Screen of the C/S Screen Dialogue to specify this organization.
			Field not used.
		BLANK	Organization indicated in the -CS screen of the Dialogue.
		V	Indexed organization.
		W	Access managed by procedural code.
4	8		EXTERNAL NAME ERROR MESSAGE FILE
			NOTE: for a TUI (or character-based) application, it is recommended to use the -CS Screen of the C/S Screen Dialogue to specify this external name.
			Field not used.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			IBM CICS - VSAM: DDNAME 8 characters.
			GCOS8-TP8: external name generated in the SELECT clause (first 4 characters); blocking factor generated in the FD clause (last 4 char.).
			Other variant: logical name generated in the SELECT clause.
5	4		FIRST SCREEN CODE OF THE DIALOGUE
			TO BE ENTERED IN THE CLIENT ONLY.
			In CICS:
			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.
			With MONIT option:
			The system generates a 'monitor' program, which calls the indicated screen at the beginning of a Dialogue. (the external names of the first displayed screen and of its corresponding program must be defined at the Screen definition level).
6	4	NUMER.	COMPLEMENTARY COMMON AREA LENGTH
			PURE NUMERIC FIELD
			Pacbench C/S : to be entered in the TUI Client only
			(C/S Screen entity).
			The length of the conversation area is determined by the combined lengths of: the screen code (6 char.), a common area data structure, and a field used to store the access keys of displayed segments, including technical information (transparent to the user).
			This last field's length is 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.
			See the 'The Generated Program' chapter in the 'On-Line Systems Development' manual specific to your environment for information on these fields.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		For BULL Gcos7 and Gcos8 and when the generation option = 'OFF' (see 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
		For the IMS cobol generation variant, this field contains the PSB code of the application in case DL1 Databases or PSB Alternate are used (required when CALLTYPE=CHNG option).
		For the HP3000 cobol generation variant, this field contains the ALLBASE/SQL Database Block code when the connection to the database is performed via the client monitor.
8	47	OPTIONS
		Pacbench C/S : to be entered in the TUI Client only (C/S Screen entity).
		If you select several options, separate each of them with a blank.
	INICHECK	Management of the presence of the Data Elements Option to be set for a TUI Client Dialogue in association with the CHECKSER or VECTPRES options entered in the General Description Screen (-GO) of the Business Component.
		In this case, a data presence buffer is generated and initialized to 'P' (absent data element).
		When transfers are performed for an update (F30 function),the host-variable indicator of each data element really transferred is set to 'N', the display transfers are also conditioned by the data element presence ('N').
		NOTE: This option can only be used if the transfer direction is "in both directions" and if the Logical View does not contain any data elements belonging to the 'after the repetitive' category.
	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 'COMMENTS' screen (-GE screen, F-type lines followed by T or C-type lines).
	MONIT	Use of a client monitor or sub-monitor. The transfer of one client to another as well as the call of servers will be done in the monitor. In the case of variant for which the monitor is required (MICROFOCUS), this option allows you to manage the calls to the server in the monitor.
	NOOCF	The first entry in the OCF program will not be positioned at 1 (deletion of the F0160 function).
	OFF	The Programs and the maps of the Dialogue will be generated with the option MDT-OFF.
	ON	The Programs and the maps of the Dialogue will be generated with the option MDT-ON (by default).
	NORPT	No automatic processing of Data Elements repeated in the Screen (validation, transfer and positioning attributes).

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	TS	CICS: Management of the communication area outside of the LINKAGE SECTION. This field is described in the WORKING and saved by default in TEMPORARY STORAGE. In case of the MDT-OFF and of the call of the HELP function the Screen is saved in the one area.
	NOSAV	TUXEDO, MICROFOCUS, GCOS7, GCOS8, HP3000: Avoids the generation of the description and of the physical access to the HE file (backup for HELP).
	NOATTR	IMS: Generation of the protected fields in the MOD without their attributes (ATTR=YES). This option is valid for all Screens of the same Dialogue.
	SERVER	MICROFOCUS WINDOWS/NT: Management of display by the multi-applications server.

Comments (-GC)

The 'Comments' screen (-GC) enables you to write and edit text in relation to one selected instance. This screen is available on all entities.

Each comments line is made up of a 60-character field containing text and a 'Type of Line' field for particular purposes (comment, call of a P.I.A, indication of an alias).

CALLING A P.I.A.

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

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

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

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

Additional lines cannot be inserted between called lines.

For more information see the 'Character Mode User Interface' guide, chapter 'Documentary Facilities', subchapter 'Parameterized Input Aids'.

PREREQUISITES

The instance must be defined prior to being documented.

The called P.I.As must also exist.

LINES COPY

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

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			On a blank-type line, you enter a free comment.
			In order to select only certain lines of the comments of an instance, use print option -EG and indicate the \$OFF command at the left end of each line to be ignored. To re-activate the -EG option on a line, indicate a \$ON command, left-justified, after the last line to be ignored.
			On an A-type line, enter the COBOL Segment name. (see chapter 'Generation of Copy Book', subchapter 'Description of VA Pac PIA 'Data'', value 'A*' in the 'Data Structure Code in Gener. Descr.' field.)
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.

Generated Constants/ Generation of a PSB (-GG)

The 'Generation Elements' screen is used to:

- modify the generated constants (which depend on the hardware in use),
- generate a PSB per sub-Monitor (as opposed to a general PSB for the whole dialogue).

MODIFICATION OF THE GENERATED CONSTANTS

To modify the generated constants, use the 'G'-type line in the 'C2' option (O: C2) (the same as when using PIA's).

>>>> For more information see the 'Character Mode User Interface' guide, chapter 'Documentary Facilities', subchapter 'Parameterized Input Aids'.

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.

The usable codes followed by the hardware code are:

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. It is possible to declare up to 10 types of terminals in the following way:

```
A LIN : T DESCRIPTION
      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).

04 ALL HARDWARE:

Modification of the external name of the Program that manages the accesses to Pactables.

(Default value: PACTABLE).

05 TUXEDO-MICROFOCUS-DPS7-DPS8:

Modification of the external name of the backup file of the Screen in case the HELP function is called. Coding: XX EXTERN where XX is the name of the file and EXTERN the external name.

(Default value: HE SAVESCR)

06 CICS MULTI-SCREENS:

Modification of the first four characters of the name of the TS used by the HELP function.

(Default value: PAC7).

12 IBM CICS, GCOS7, GCOS8, MICROFOCUS, TUXEDO:

Modification of the name of the sub-program called for the reception and the update of the message.

(Default value: ZAR980)

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

16 CICS:

Possibility of writing the TEMPORARY STORAGE in AUXILIARY STORAGE (AUXILIARY) instead of MAIN STORAGE.

(Default value: MAIN).

21 IBM CICS:

This parameter takes the ARRAY or TABLE values if in the Screen there is a horizontally and vertically repeated field. The logical message will be initialized from the physical message which keeps it in phase.

22 IBM CICS:

For the multi-screen variant, the name of the subprogram destined to replace the standard PRCGI subprogram (external name D4R980) that formats a 3270-type message.

IMS: GENERATION OF A PSB PER SUB-MONITOR

To generate one PSB per sub-monitor, indicate on a G-type line in the 'Generation Elements' screen (-GG) of each sub-monitor:

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

This line must be the first -GG line.

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

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

Setting Generation Options (-GO)

The coding of an option line is done on an O-type line in the 'Generation Options' screen (-GO).

The coding of these options is:

OPTION=value of the option (no space before or after the = sign).

Several options can be indicated on the same line, separated by at least a blank.

No check is performed on the input of the options but if the options entered do not correspond to the standard ones, a message appears on the generation report.

All the options are entered preferably at the Dialogue level to ensure consistent generation of the components attached to the Dialogue.

The table below lists the required options. It shows in which components you must enter a value or another in order to generate some component or other. For example, to generate a Client component, you will have to set options also at the Server (Dialogue) level.

OPTIONS SUMMARY TABLE: TUI CLIENT AND MONITORS

This table reads as follows :

- the first column lists the options, in alphabetical order,
- the other columns indicate in which component they must be present to generate the component which is first in the column.

ABBREVIATIONS USED

SVR MON Server Monitor (MS type)
 CLI C/S Screen (C type)
 CLI MON Client Monitor (SC type)
 SRV Business Component (S type)

! saisir les options	! Pour générer !		
!	! MON SRV	! CLI	! MON CLI
! ACCESERR	!	! SRV	!
! CALLSPG	! MON SRV	!	! MON CLI
! CALLTYPE	! MON SRV	!	! MON CLI
! CHANGE	!	! SRV	!
! CHECKSER	!	! SRV	!
! DATAERR	!	! SRV	!
! FORMAT	!	! CLI	!
! INITSERV	!	! CLI*	!
! LGCOMM	!	! SRV*	!
! LGCOMZS	! MON SRV	!	! MON CLI
! LTHENATT	!	! CLI	!
! LTHREPET	!	! CLI	!
! LTHSCREEN	!	! CLI	!
! MONSER	!	! SRV	!
! NUVERS	!	! CLI	!
! TERMSERV	!	! CLI*	!
! TRAN	!	! SRV	!
! VECTPRES	!	! SRV	!

*: Input required (no default value)

>>>> Each of these options is described in details in the following pages.

OPTIONS USE

>>>> To know precisely the use context for an option, see the summary table in the previous page.

- ACCESERR=nnn

Default: 001.

Number of errors on the Segment that a server can find (must be 3 characters long).

- CALLSPG=CALL (Gcos 8)

LINK ((Gcos8) same as CALL ILINK

Default: CALL

Call type used by the client or server monitors to call the client or server components associated with the monitors.

If there is no client monitor, it is the call type of the server component by the client component.

For Gcos8 , you must specify the same value in the CALLSPG option and in the CALLTYPE option.

- CALLTYPE=LINK (CICS)

CALL

Default: LINK.

Server call type for CICS client to CICS server.

- CALLTYPE=CHNG (IMS) same as CALL CHANGE

CALL

Default: CALL.

Server call type for IMS client to IMS server.

A PSB (with alternate PCB) must be indicated on the -O of the server Dialogue for the CALLTYPE=CHNG option.

- CALLTYPE=NTPR (Gcos7-Gcos8) same as NEXT TPR

CALL (Gcos7-Gcos8)

LINK (Gcos8) same as CALL ILINK

Default: CALL (Gcos8), LINK (Gcos8).

server monitor call type for the client monitor.

You must specify the same CALLTYPE value for the client and for the server.

- CHANGE=YES or NO

Default: NO

For TUXEDO only

Processes the TPNOCHANGE parameter (if CHANGE=NO) or the TPCHANGE parameter (if CHANGE=YES) at the time of the CALL.

- CHECKSER=YES or NO

Default: NO

Check of the Logical View data performed by the server component.

This option also allows conditioning the transfer of the Logical View data on to the SQL segments according to the Data Element presence.

No check is generated on the occurred elementary Data Elements.

CAUTION: This option can only be used if the transfer direction is "in both directions" and if the Logical View does not contain any data elements belonging to the 'after the repetitive' category.

- DATAERR=nn

Default: 01.

Number of errors on the Logical View data that a server can find (must be 2 characters long).

- FORMAT: INTERNAL
 EXTENDED

Default: INTERNAL.

With the INTERNAL option, the Data Elements of the Logical View are generated using the internal format and take into account the usage of this Data Element.

With the EXTENDED option, the format of the Data Elements is recalculated in the following way: the indicated usage is ignored and replaced by the DISPLAY use, and if the Data Element is a signed numerical Data Element, the 'SIGN IS LEADING' usage replaces the indicated usage.

- INITSERV=xxinit

Call of an initialization server.

xxinit corresponds to the occurrence code of this server.

- LGCOMM=nnnnn

Length of the communication area between the server and the Communications Monitor or between the client and the server or between the client monitor and the server.

The value entered must be 5 characters long.

- LGCOMZS=nnnnn

Default: 32000

Length of the communication area between the client and server monitors.

This option must be entered at the client and server monitors.

The value entered must be 5 characters long.

To parameterize the length of the backup areas in the linkage section of a client component in the case of a server call by the monitor (MONIT option entered in the client Dialogue in the -O screen), you can use the following three options:

- LTHENATT=xxxx

Default: 2000

Length of the backup area of the DE-ATT Table.

- LTHREPET=xxxx

Default: 1000

Length of the backup area of the repetitive.

- LTHSCREEN=xxxx Code

Default: 6000

Length of the backup area of the screen message.

- MONSER=occurrence code of the server monitor

Default: no server monitor.

- NUVERS=xxxxx

Default: generation session number of the server and the TUI client.

This option allows transfer to the server, via the technological buffer, of the data required to process the different versions of client or server components.

- TERMSERV=xxterm

Call of the termination server.

xxterm corresponds to the occurrence code of this server.

- TRAN=YES or NO

Default: YES

This option allows you to use or not the transactional mode for the TUXEDO variant.

- VECTPRES=YES or NO

Default: NO

This option allows management of the presence of the logical view's Data Elements.

If this option is set to YES, a data presence buffer is generated each elementary data element of the view in the communication area between the client and the server. This buffer determines the presence of the data element and thus conditions the loading of the host-variable indicators for the SQL tables.

This option is already integrated in the CHECKSER=YES option, which needs the same buffer for the data checks that may not be desired.

Setting VECTPRES to YES is therefore not necessary if CHECKSER=YES. In a character-based client, as for the CHECKSER option, the positioning of the data

element absence (value 'P') in the associated buffer must be written in structured code, the default being always 'data element present'.

CAUTION: It can only be used if the transfer direction is 'in both directions' and if the Logical View does not contain any data elements belonging to the 'after the repetitive' category.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
4	60		DESCRIPTION
			If no P.I.A. is called, you can use the whole field.
			On an O-type line, Enter the option followed by an equal sign and the value (with no space).
5	40		DESCRIPTION / SECOND PART
			This field is specific to a P.I.A. call.
			With value 'C2' in the OPERATION CODE field, the cursor automatically tabs to the first position of this field.
			This field is initialized with 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.

Writing Procedural Code (-P)

Introduction

>>>> For complete information on this subchapter, the following readings are recommended:

the 'Structured Code' manual for details on the syntax used to write specific procedures,

the 'Pacbench C/S - Business Logic' manual for details on the insertion rules in the different services.

Operators Used by Pacbench Client/Server

This section presents a selection of operators often used with Pacbench C/S.

To consult the complete list of all operators, refer to the 'Structured Code' manual, chapter 'Modifying the Procedure Division', subchapter 'Procedural Code Screen' section beginning with 'Operators'.

SEGMENT ACCESS OPERATORS

Xaa: Followed by a Segment code in the OPERAND, generates a PERFORM of label F80-ffnn-aa.

If the Segment code is followed by an extraction method (after a space), generates a PERFORM of F80-ffnn-aa-code of extraction method (Server components only).

Yaa: Generates a COBOL paragraph label for Function 80. Followed by a Segment code in the OPERAND field, generates a F80-ffnn-aa COBOL paragraph label (used when the physical access to a Segment is replaced).

For these operators, 'aa' is a generated code.

The values of this code are documented in the 'Pacbench C/S - Business Logic' manual.

OPERATORS FOR ERROR POSITIONING

Specific errors are coded in Procedural Code lines (-P). The corresponding error messages are entered according to rules specified in subchapter 'User-Defined Error Messages', Chapter 'Error Messages - Help Function'.

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

ERU: User-related error.

Operand:

1 to 4: Error code (user-defined)

5: Blank

after 5: Code of the variable Data Element (optional: ensures the cursor positioning).

The ERU operator cannot be used on repetitive Data Elements.

ERR: Error on Data Element.

Operand:

1: Code of the error (alphanumeric character except for '0' and '1' which are reserved for coding documentary messages)

2: Blank

after 2: 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 Data Element in a repetitive category, its code must be entered and has to be followed by the sequence number of the Data Element occurrence.

EXAMPLE: DELCO (2)

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, as described on the definition screen of the client component. If no code is entered, the message will be blank.

OPERATORS FOR END-OF-PROCESSING

GFA: End of display processing.

(Branch to 'END-OF-DISPLAY' label)

GFR: End of reception processing.

(Branch to 'END-OF-RECEPTION' label)

GDB: Return to the beginning of current iteration.

GDI: Go to the beginning of next iteration.

GFT: Go to the end of the iteration.

(Branch to the end of category in 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).

NOTE: 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 line following a branching operator. Indeed, this will trigger a compiling error in the generated COBOL.

OPERATORS FOR SCREEN BRANCHING

OTP: Call of the Screen whose external name is indicated as the OPERAND.

OSC: Call of the Screen indicated as the OPERAND.

OSD: Call of the Screen indicated as the OPERAND (the end of the reception processing).

OPERATORS FOR DATE PROCESSING

AD6: Transforms the system date into a 6-character date format of 'MMDDYY' or 'DDMMYY' depending on the value in the DATE FORMAT IN GENERATED PROGRAMS field specified on the Library Definition screen. The transformed date is moved into the Data Element indicated in the OPERAND field.

AD8: Transforms the system date into an 8-character date with slashes, 'MM/DD/YY' or 'DD/MM/YY' depending on the value in the DATE FORMAT IN GENERATED PROGRAMS field on the Library Definition screen. The transformed date is moved into the Data Element indicated in the OPERAND field.

AD: Transforms the date according to the format types specified in the OPERAND. A date may be formatted in the following ways :

```
+-----+-----+
! CODE ! GENERATED FORMAT !
+-----+-----+
! I   ! 'Internal' : YYMMDD   !
!-----!-----!
! D   ! 'Display' : MMDDYY or DDMMYY !
!     ! according to the value entered in the DATE FORMAT !
!     ! IN GENERATED PROGRAMS field on the Library Def. !
!     ! screen. !
!-----!-----!
! E   ! 'Extended' : MM/DD/YY or DD/MM/YY !
!     ! according to the value entered in the DATE FORMAT !
!     ! IN GENERATED PROGRAMS field on the Library Def. !
!     ! screen. !
```



```

!-----+-----!
! S ! 'Internal' : CCYYMMDD !
!-----+-----!
! C ! 'Display' : DDMCCYY or MMDDCCYY !
! ! according to the value entered in the DATE FORMAT !
! ! IN GENERATED PROGRAMS field on the Library Def. !
! ! screen. !
!-----+-----!
! M ! 'Extended' : DD/MM/CCYY or MM/DD/CCYY !
! ! according to the value entered in the DATE FORMAT !
! ! IN GENERATED PROGRAMS field on the Library Def. !
! ! screen. !
+-----+-----+
! G ! 'Extended' : CCYY/MM/DD !
+-----+-----+

```

The different formats can be set at the Data Element input, internal or output format level or at the conversational format level.

NOTE: When several lines are required to enter the condition of a date processing operator, the operands must be written on the line following the last one and the condition.

EXAMPLE: In order to change an 'I'-formatted date into a 'D'-formatted date, 'AD' should be entered in the OPERATOR field and 'ID DELCO1 DELCO2' in the OPERAND field; DELCO1 being the Data Element containing a YY/MM/DD date format (the Pacbase constant DATOR may be used) and DELCO2 being the Data Element containing the changed date format: DD/MM/YY or MM/DD/YY.

```

A SF LIN OPE OPERAND          LVTY CONDITION
BB 100                          99IT COND1
BB 110                          OR COND2
BB 120 AD ID DELCO1 DELCO2

```

AD0: Century positioned from the DAT-CTY field initialized to '19' and that can be modified by the user.

AD1: Century set to '19' if System year is less than the value in DAT-CTYT field (default = '61'), or to 20 otherwise.

AD2: Century set to '20' if System year is less than the value in DAT-CTYT field (default = '61'), or to 19 otherwise.

OPERATORS FOR TIME PROCESSING

TIM: Hour display in 'HHMMSS' format from the EIBTIME field under CICS, from the TIME field with other hardware.

Example:

```

A SF LIN OPE OPERAND          LVTY CONDITION
BB 100 TIM FIELD1

```

TIF: 'HHMMSS' format changed into 'HH:MM:SS'.

Example:

```

A SF LIN OPE OPERAND          LVTY CONDITION
BB 100 TIF FIELD1 FIELD2

```

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 also used 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:

This function contains a sub-function for each updatable field on the end-user system. The sub-function into which the procedural code is to be inserted is identified by the code of the corresponding Data Element.

Enter the Data Element code, left-justified, in the CONDITION FOR EXECUTION field.

In the TY field, enter the type of block:

***A:** to INSERT the sub-function before the automatic sub-function.

***P:** to INSERT 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.

FUNCTIONS F25, F35 and F60

These three functions contain a sub-function for each accessed Segment. The sub-function into which the procedural code is 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.

In the CONDITION FOR EXECUTION field, enter the Segment code, left-justified, and possibly the category code:

Screen-top = ' ', Repetitive = 'R', Screen-bottom = 'Z'

Chapter 6. Monitors

TUI Applications Monitors / Sub-Monitors

INTRODUCTION

Two basic types of architecture are possible, one which uses a monitor program, and the other which does not.

In the architecture without monitor, the Client components communicate directly with the Server components.

However, some requirements of the application (confidentiality, data encryption) or technical constraints (communication protocols) will require the use of an architecture with a monitor program.

The monitor option allows you to interface more easily with your own communication method and to insert data security and encryption/decryption processing.

MONITORS / SUB-MONITORS

Client Monitor :

- initializes the conversation,
- ensures the link between the different Client components,
- performs the call of the service requested or of the Server Monitor corresponding to this service.

Server Monitor :

- receives the information sent by the client using the Communication Area,
- calls the server corresponding to the service requested by the client,
- returns the information to the Client Monitor

A sub-monitor ensures :

- the link between the client components of this sub- monitor,
- the link to another sub-monitor for the call of a Client component which does not belong to this sub-monitor,
- the call of the requested service or of the server Monitor corresponding to this service.

Client sub-monitor :

Each Client component that depends on a Client Monitor can be defined in that Client sub-monitor. A sub-monitor is therefore a group of Client components whose choice can depend on logical considerations (Client components working in the same domain) or system (division in function of tasks: consultation or updates, running priority ...).

The use of sub-monitors and of the list of Client components (that make up the sub-monitors) is determined by the WORKING AREA screen (-W).

```

-----
!
!WORK AREA OF THE FLIGHT RESERVATION SCREEN O ZR (CLIENT) !
!
!BEGINNING LINE NUMBER : AA !
!
! A LIN S LEVEL          DESCRIPTION !
!
!   100 M ZRMON1        !
!   110 C ZR00ME        !
!   120 C ZR00RE        !
!   130 C ZR00VO        !
!   140 C ZR00CV        !
!   200 M ZRMON2        !
!   210 C ZR77PI        !
!   220 C ZR77AV        !
!   230 C ZR00ME        !
!   240 C ZR77DE        !
!
-----

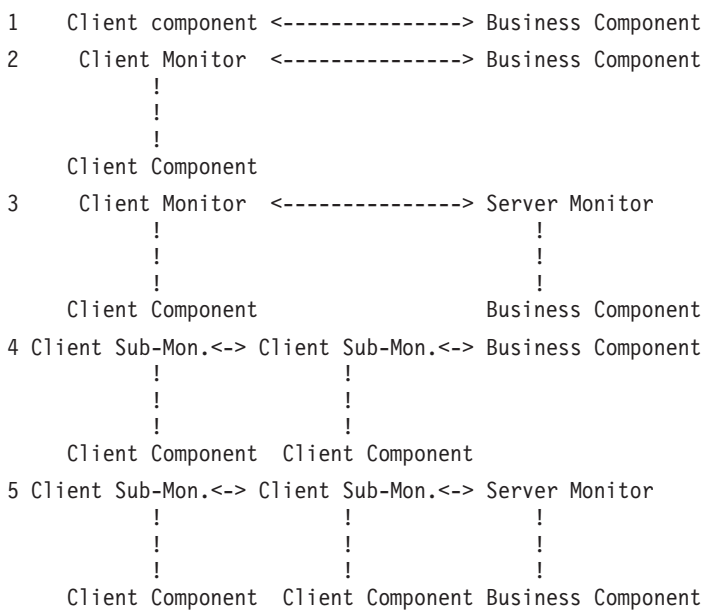
```

In this example, two sub-monitors are generated: ZRMON1 and ZRMON2 (lines with an 'M' type declaration).

The screens belonging to each sub-monitor must follow the declaration line of the sub-monitor with a 'C' type declaration. Each screen must belong to one of the sub-monitors.

POSSIBLE ARCHITECTURES FOR A TUI APPLICATION

The possible architectures are the following :



NOTE: The communication type depends on the generation variant of the Client and Business Components.

Graphic Application : Communications Monitor

This subchapter explains the development of a VisualAge graphic application using the standard or single-view mode.

>>>> For a complete documentation on the Communications Monitor, refer to the 'Pacbench C/S - Business logic' manual.

```

+-----+
!                               ENGLISH DOCUMENTATION                               PDSG.NDOC.ACC.439   !
!                               1 2                                               !
! MONITOR CODE                   CLCFOL                                           !
!                               !                                                 !
! MONITOR NAME .....: Communic. Monitor Client Fold3r                          !
!                               !                                                 !
! MONITOR TYPE .....: FM 4   FOLDER MONITOR                                     !
! MESSAGE LENGTH * 1024 CHARACTERS...: * 08 5                                   !
!                               !                                                 !
!                               !                                                 !
!                               !                                                 !
! SEND OF ERROR LABELS.....: * Y 6                                              !
!                               !                                                 !
!                               !                                                 !
! TYPE OF COBOL AND MAP TO GENERATE...: 7 3 8 1   OS/2 MICROFOCUS                !
! CONTROL CARD OPTIONS FRONT & BACK...: 9 10                                     !
! EXTERNAL NAMES .....: 11CLCOMM   COMM. TYPE: 12LOCAL                          !
! TRANSACTION CODE.....:13                                                       !
!                               !                                                 !
! EXPLICIT KEYWORDS...: 14                                                       !
! UPDATED BY.....:           ON :           AT:   :   LIB : ACC                   !
! SESSION NUMBER ....: 0377   LIBRARY: ACC   LOCK:                               !
!                               !                                                 !
! O: C1 CH: 0 clcfol                   ACTION:                                   !
+-----+

```

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		COMPLETE CODE
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	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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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.
5	2		MESSAGE SIZE * 1024 CHARACTERS
			The message size is given in K bytes. It must be between 2 KB and 32 KB. The default value is 24 KB.
6	1		SENDING ERROR MESSAGES
		BLANK	Retrieval of the value entered at the higher level.
		Y	Error messages sent. (Default value at Dialogue level)
		N	No error messages sent. Only the key of an error message is sent.
7	1		TYPE OF COBOL TO GENERATE
		N	No generation
			ON-LINE SYSTEMS DEVELOPMENT FUNCTION:
		X	IBM MVS COBOL II
		O	IBM MVS COBOL/VS
		X	IBM VSE COBOL II
		1	IBM VSE COBOL/VS
		3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT COBOL
		O	IBM AS/400 COBOL
		4	BULL GCOS7 COBOL
		5	BULL GCOS8 COBOL

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	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 OR VSE 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).
	3	MICROFOCUS COBOL, IBM AIX-OS/2-WINDOWS/NT 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
8	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
	0	Variants 0, 1, X for CICS (IBM) (programs and BMS maps).
		Variant 2 for IBM 36, monochrome map.
		Variant 3 for PC/MICROFOCUS (MS/DOS).
		Variants 4, 5, 6 for QUESTAR.
		Variant I for DEC/VAX.
		Variant U for UNISYS 2200 (programs and FLDP maps).
		Variant 8 for UNISYS Series A SDF format.
		Variant Y for IBM 38.
		Variant O for AS 400.
		Variant M for GCOS6-DTF-DFC-Vision.
		Variant 7 for VPLUS.

NUMLEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
	1	Variants 0, X for IMS (IBM) (programs and MFS maps).
		IMS (IBM) (programs and MFS maps).
		Variant 2 for IBM 36, map in color.
		Variant 3 for PC/MICROFOCUS (OS2).
		Variants 4, 5, 6 for VIP TYPE.
	2	Variants 0, 1, X for CICS (IBM), BMS map in color.
		Variant 3 for MICROFOCUS (UNIX).
	3	Variants 0, X for IMS (IBM), MFS map in color.
	4	Variants 0, 1, X for IMS (IBM) monitor.
		Variant 3 for IBM VISUALAGE COBOL program.
	5	Variants 0, 1, X for CICS (IBM) monitor.
	C	Variants 4, 5, 6, U for MULTI-TERMINAL.
		Variants 0, 1, X for CICS MULTI-TERMINAL.
		Variant K for ICL.
		Variant 8 for UNISYS Series A Logical Screen.
	F	Variant 4 for TDS FORMS (BULL GCOS7).
		Variant M for DM6 TP FORMS.
		Variant R for 'TPSVRINIT service' type Program or 'TPSVRDONE service' type Program.
	R	Variant F for 'REQUESTER' type Program. or variant R for 'CLIENT' type program.
	S	Variant F for 'SERVER' type program. or variant R for 'SERVICE' type program.
		PACBENCH CLIENT/SERVER
		TP MONITOR
	0	Variants 0, X for CICS (IBM) (Program and BMS map for the client).
		Variant 3 for MICROFOCUS MS/DOS.
		Variant F for TANDEM Pathway.
		Variant R for TUXEDO.
		Variant 7 for VPLUS.
		Variant U for UNISYS-2200.
		Variant 8 for UNISYS-A.
	1	Variants 0, X for IMS.
		(program and MFS map for the client).
		Variant 3 for MICROFOCUS OS/2.
		Variant F for TANDEM TUXEDO.
		Variant 8 for UNISYS-A Open/OLTP.
	C	Variants 0, 1 for MULTI-SCREEN CICS or variant X for CLIENT ONLY.
		Variant 4 for MULTI-SCREEN GCOS7.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Variant 5 for MULTI-SCREEN GCOS8.
			Variant R for TUXEDO (from release 6.2 and TUXEDO (from release 6.2 and higher).
		2	Variants 0, X for CICS (IBM) (program and BMS color map).
			Variant 3 for MICROFOCUS UNIX.
		3	Variants 0, X for IMS (program and MFS color map).
			Variant 3 for MICROFOCUS.
		4	Variant 3 for IBM VISUALAGE COBOL.
9	8		EXTERNAL NAME or CLASS PREFIX
			For a Folder or a Communications Monitor, this field must contain its external name.
			For a Folder View, this field contains the class prefix.
			Default value: the occurrence code.
			For a Folder extraction via the GVC command, the value recognized is the occurrence code even if another value was entered in the field.
			In a Folder View, this field contains the class prefix. Value required in this field.
			NOTE: The external name given with the extraction of a Folder View - via the GVC command - is the occurrence code of the Folder View.
10	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 '\$' is used to disable the Business Component generation and to enable the Services Manager generation.
11	1		CONTROL CARDS AFTER PROGRAMS
			Option code which selects the JCL lines to be inserted after each generated program.
		\$	No generation of program.
12	8		TYPE OF COMMUNICATION
			The type of communication is required. The choice on the type of communication depends on the generated COBOL variant and possibly on the TP Monitor variant.
			>>>> Refer to the summary table of compatibilities in the 'Concepts and Architectures' manual, chapter 'Execution Environments', in the Developer's documentation dedicated to eBusiness Applications.
		ECINOEXT	ECI no-extend
		CPIC	SNACPIC or MSCPIC
		XA	TUXEDO in transactional mode
		NONXA	TUXEDO in non-transactional mode

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		SOCKET	TCP-IP
		TCIS	TCIS
		MQSERIES	MQSERIES
		XCP2	CPIC-XCP2
		LOCAL	Local communication : default value.
		TCPTDS	TCP/IP Access TDS
13	8		MONITOR TRANSACTION CODE
			This field is to be filled in for the following platforms:
			. CICS,
			. IMS,
			. UNISYS 2200.
14	55		EXPLICIT KEYWORDS
			This field allows you to enter additional (explicit) keywords. By default, keywords are generated from the instance's name (implicit keywords).
			Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage and are therefore ignored in keywords.
			Keywords are not case-sensitive: uppercase and lower-case letters are equivalent.
			NOTE: 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'.

Chapter 7. Error Messages - Help Function (TUI)

Error Messages: Introduction

You can manage error messages used to inform the end-user of input errors detected by the 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 GEC print-generation command. It generates the error messages for the screens specified in the GEC command in the PAC7GK file. Error messages of other screens found in the PAC7LK file are copied without being modified in the PAC7GK file.

AUTOMATIC ERROR MESSAGES

An error message record is automatically generated for each validation instruction entered on the Screen Call of Elements or for each file access.

Error messages are divided into two parts:

- A message corresponding to the error type and thus, 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 4 error messages for various validations on Data Elements:

```
'INVALID ABSENCE OF DATA ELEMENT  
'INVALID PRESENCE OF DATA ELEMENT  
'NON-NUMERICAL/NON-ALPHABETICAL CLASS DATA ELEMENT  
'INVALID VALUE FOR DATA ELEMENT
```

SEGMENTS

The System generates the following standard file access error messages:

.Segments used for input:

```
'INVALID CREATION' and 'INVALID MODIFICATION/DELETION.
```

Both messages include the Segment's name.

.Segments used for display:

```
'END OF LIST'
```

```
'ABSENCE OF RECORD'
```

These automatically generated error messages may be replaced by specific messages. For more details, refer to sections 'Overriding Automatic Error Messages' - 'Business Component' and 'TUI Clients').

USER-DEFINED ERROR MESSAGES

Only the validations coded on the description lines of a Screen and the file accesses trigger the automatic generation of error messages.

For all types of errors detected by other validations, the user-defined error messages must be explicitly declared.

They are declared on the 'Error Messages-Help' screen (-GE), via the Description lines of either the Screen or the Dialogue according to the type of error.

Errors on the data or the Logical View can be detected by a server.

These error messages must be explicitly declared on the 'Error Messages-Help' screen, via the Description lines assigned to the call of the Data Element in the Logical View.

Business Component:Overriding Automatic Error M.

1. MESSAGE ASSOCIATED WITH A DATA ELEMENT

You can override an error message associated with a Logical View Data Element via the 'Error Messages-Help' screen of the line that calls the Data Element in the Logical View (CH: S...CEnnnG).

NOTE: You can override the message directly in the 'Error Messages-Help' screen of the Data Element (CH: E.....GE).

In this case, the replacement affects all the logical Views in which this Data Element is called, except for the Logical Views where the replacement for this Data Element is made on the 'Error Messages-Help' screen associated with their Data Element call line. (CH: S...CEnnnGE).

- Type = U (error message).
 - 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 (limited to 55 characters).

2. MESSAGE ASSOCIATED WITH A SEGMENT

The error message associated with the processing of accesses on Database Segments is replaced on the 'Error Messages-Help' screen attached to the Business Component (CH: O.....GE).

NOTE: You can override the message directly on the Physical Segment 'Error Messages-Help' screen (CH: S...GE).

In this case, the replacement affects all the Business Components in which the Segment is called except for the Business Components where the replacement for this Segment is made on the 'Error Messages-Help' screen.

- Type = S (Segment call)

You do not have to enter this line in the case described in the 'NOTE' above (replacement in CH: S....GE).

1 to 4 : Segment code
5 : blank or W (possibility to associate a WARNING with the error message)
6 : category code (required if the same segment code is used in several categories):
Screen-top : ' ',
Repetitive : 'R',
Screen-bottom : 'Z'
7 : Segment position (1,2,...) in the category (required if the same Segment code is used several times in a category).
.TYPE OF LINE value = 'U' (error message)
1 to 4 : . Use in reception:
'NFND' record not found,
invalid modification/deletion
'DUPL' invalid creation.
. Use in display:
'END' end of list.
'ABSC' absence of record
5 : blank or W (possibility to associate a WARNING with the error message)
6 : error message (limited to 55 characters)

TUI Client: Overriding Automatic Error Messages

You can override the error message on the 'error Messages-Help' screen associated with the C/S Screen (CH: O.....GE).

You can proceed differently for Data Elements, see below.

Enter the following lines :

.An 'F' or 'S' type of line value for Data Element call

or Segment call, respectively.

.An 'U' Type of line value for error message.

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

NOTE: The error messages are also displayed as online help.

- You can add, in the screen CH: O.....GE, a Comments field or a text call, displayed in online help only. For more details, see the following subchapter which is dedicated to HELP FUNCTION.

1) MESSAGE ASSOCIATED WITH A DATA ELEMENT

NOTE: You can override an automatic error message in the 'Error Messages-Help' screen of the Data Element (CH: E.....GE).

In this case, the replacement affects all the Screens in which this Data Element is called, except for the Screens where the replacement for this Data Element is made on their 'Error Messages-Help' screen. Note that messages entered in this screen will not appear in the online Help.

.Type of line value : 'F' (Data Element call)

You do not have to enter this line in the case described in the 'NOTE' above (replacement in CH: E.....GE).

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 or W (possibility to associate a WARNING with the error message)

6 : error message (limited to 55 characters)

EXAMPLE:

```
F NUCLI
U 2 CLIENT NUMBER IS REQUIRED
U 5 CLIENT NUMBER MUST BE > 10001
```

2) MESSAGE ASSOCIATED WITH A SEGMENT

.Type of line value : 'S' (Segment call)

Positions :

1 to 4 : Segment code

5 : blank

6 : category code (required if the same segment code is used in several categories):

Screen-top : ' ',

Repetitive : 'R',

Screen-bottom : 'Z'

7 : Segment position (1,2,...) in the category (required if the same Segment code is used several times in one same category).

.Type of line value : 'U' (error message)

Positions :

1 to 4 : .Use in reception:

'NFND' record not found,
invalid modification/deletion

'DUPL' invalid creation.

.Use in display:

'END' end of list

5 : blank or W (possibility to associate a WARNING with the error message)

6 : error message (limited to 55 characters)

EXAMPLE :

S CD10
U DUPL PRODUCT REFERENCE NUMBER ALREADY ASSIGNED
U END END OF SEARCH FOR THIS DEPARTMENT

User-Defined Error Messages

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

- Errors on Data Elements. These are errors detected by procedural code and which can be associated with a Data Element of the C/S Screen or the Logical View.
- User-related errors. These are errors detected by procedural code and which can be common to a number of Screens of the TUI application and/or to several Business Components of the application (graphic or TUI).

ERROR ON A DATA ELEMENT

An error on Data Element can be described at different levels:

1. The 'Error Messages-Help' screen associated with the C/S Screen (CH: O.....GE),
2. The call of Logical View Data Element screen (CH: S...CE, G lines for Data Element),
3. The 'Error Messages-Help' screen of the Data Element (CH: E.....GE).

.Type of line value : 'F' (Data Element call)
(only for the input case presented in point 1).

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 or W (possibility to associate
a WARNING with the error message)

6: error message.

Several 'U'-type lines may follow an 'F'-type line.

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

- In the input case corresponding to point 1., you can add a Comment or a Text call, displayed in the online help only. For more details, see the following subchapter dedicated to the Help Function.

USER-RELATED ERRORS

A user-related error can be described at two levels:

1. The 'Error Messages-Help' screen associated with the C/S Screens Dialogue (CH: O..GE),
2. The 'Error Messages-Help' screen associated with the Business Components (CH: O.....GE).

Type of line value : 'U' (error message)

Positions:

- 1 to 4: code of the error (mentioned on a '-P' line).
- 5: blank or W (possibility to associate a WARNING with the error message)
- 6: error message

EXAMPLE:

U AV01 DATE AND NATURE OF NOTICE MUST CORRESPOND

POSITIONING USER-RELATED ERRORS

To position a user-related error, refer to :

- chapter 'Business Component, subchapter 'Writing Procedural Code', section 'Operators Used by Pacbench Client/Server', paragraph 'Operators for Error Positioning'.
- chapter TUI Client, subchapter 'Writing Procedural Code', section 'Operators Used by Pacbench Client/ Server', paragraph 'Operators for Error Positioning'.

Help Function (TUI Applications)

Help Function: Introduction

NOTE: This function is not available for graphic applications.

The HELP Function allows you to branch to a Help Documentation screen at any time.

Help Documentation has two levels:

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

You can branch to a HELP screen by positioning :

- the cursor on the appropriate field and pressing a PFkey,
- a designated character in the CHOICE field for screen-level Help,
- a designated character 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,
- Go back n pages,
- Continue scrolling through all HELP documentation, next page or the next n+1 pages after consultation of a previous page (for example, if you consult up to page 5 and then re-consult a page before page 5, you can consult page 6 with n+1),
- Stop the HELP display and return to the initial work screen; possibility to bring a coding value to the field from which the HELP was called.

To do so, the end user will indicate in the 'line number' input field the line number corresponding to the value he wants to be returned.

- To go back or forward several pages, another field allows you to input the number of pages.

The Definition screen of the HELP screen must be created, but you do not have to describe it since its format is fixed and cannot be modified by the user.

The code of this screen is XXHELP, where XX is the Dialogue Code. The external name must also be indicated ('PACHELP' by default). The Screen must then be generated using the appropriate variant.

As for error messages, the content of online help is also generated on request in the error message sequential file.

This file is used to create the applications' error message files.

These are indexed files or databases depending on the hardware in use.

The Help Function displays the documentation which is taken from this file.

Help Messages : Coding

SCREEN - RELATED HELP DOCUMENTATION

Screen-Help Documentation is made up of information entered in the 'Error Messages-Help' screen associated with the C/S Screen considered (CH: O.....GE).

The content of this help is generated from the following lines, in the following order :

- Comments or text calls on lines of the type:

C: Comments

Positions:

1 to 5: blank

6: Comments

T: : Text Call

Positions:

1 to 5: blank

6 to 11: code of the text called

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

>>>> See below details about the processing of texts displayed in the online help.

NOTE: These lines must be entered at the beginning, before any other types of lines.

- Error message associated with error on Segment:
Presence of two lines of the following type:

S: Segment call

For more details on these types of lines, refer to the previous subchapter dedicated to error messages.

- Comments associated with an error on Segment:
Presence of three lines of the following type:

S: Segment call

Positions:
1 to 5: blank
6: Comments

- Call of a text associated with an error on Segment:
Presence of three lines of the following type:

S: Segment call

Positions:
1 to 5: blank
6 to 11: code of the text called
12 to 13: code of the selected paragraph or '**'
to select all paragraphs.

>>>> See below details about the processing of text displayed in the online help.

INPUT FIELD - RELATED HELP DOCUMENTATION

Help Documentation associated with an Input field is generated from different sources, in the following order:

1. Information entered in the 'Error Messages-Help' screen associated with the C/S Screen considered.

- Error message associated with an error on the Data Element corresponding to the field in which data may be entered.

Presence of two lines of the following type:

F: Data Element call

U: Error message

>>>> For more details on these types of lines, refer to the previous subchapter dedicated to error messages.

- Comments associated with an error on Segment:

Presence of three lines of the following type:

F: Data Element call

U: Error message

T: Text call

Positions:
1 to 5: blank
6: Comments

- Call of a text associated with an error on Data Element:
Presence of three lines of the following type:

F: Data Element call

U: Error message

T: Text call

Positions:
1 to 5: blank

6 to 11: code of the text called
12 to 13: code of the selected paragraph or '**'
to select all paragraphs in their code
alphabetical order.

>>>> See below details about the processing of texts displayed in the online help.

2. Data Element Description lines (CH: E.....D)

In the Help Function, lines corresponding to value coding are preceded by an number of order which allows the end user to ask for the transfer of the corresponding value in the field concerned.

PROCESSING OF TEXTS DISPLAYED IN THE HELP DOCUMENTATION

Texts called in the -GE screen are aligned in a standard printing layout.

All the texts called are justified if they contain a \$n-type parameter. If not, they are displayed as such. However, if a non-justified text follows a text containing justification, it has the same justification as the previous text except if it contains a \$\$n parameter.

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

Line skips are specified in the text, but limited to a three-line skip maximum (line skips greater than three become 3); page skips become three-line skips.

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

Thus, the same text can be a part of the 'On-Line' Help Documentation and of an application User's Manual.

For complete information on text processing in relation to the Help Function, see the 'Personalized Documentation Manager (PDM)' manual, chapter 'Structure and Contents of a Document', subchapter 'Print Options', section 'Text-Processing Options'.

Coding Screen (-GE)

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

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

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

Error Messages & On-Line Help Generation/Printing

GEC: Generation and printing of error messages.

C1: Generation of error messages defined for the Client or Server Dialogue and for each component.

C2: Generation obtained through option 1 plus generation of the On-Line Help Documentation.

C3: Generation of error messages for the Client Dialogue only.

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 (set in the REST procedure) is used by default.

LEC: List the error messages defined for the Dialogue and for each Screen.

This list only includes messages that have already been generated.

NOTE: If a Screen suffix is entered on the Continuation Line of a GEC or LEC 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.

- In TP mode, 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.

Example 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 lines can be associated with a given screen.

```

+-----+
!           ENGLISH DOCUMENTATION LIBRARY           *PDMC.VDOC.ACC.3!
! ON-LINE SCREEN CODE           ZRHELP           !
!
! SCREEN NAME.....: HELP           !
!
! SCREEN TYPE.....: C           C/S SCREEN           !
! SCREEN SIZE (LINES, COLUMNS) .....: 24           080           !
! LABEL TYPE, TABS, INITIALIZATION...: L           01           -           !
! HELP CHARACTER SCREEN, DATA ELEMENT: 10           11           !
!
!
!                                LABELS  DISPLAY  INPUT  ER.MESS.  ER.FL!
! INTENSITY ATTRIBUTE .....: N           N           N           N           N !
! PRESENTATION ATTRIBUTE .....: N           N           N           N           N !
! COLOR ATTRIBUTE .....: W           W           W           W           W !
!
! TYPE OF COBOL AND MAP TO GENERATE..: 0 1           IBM OS IMS (PROG. & FOR. MFS!
! CONTROL CARD OPTIONS FRONT & BACK..:           (PROGRAM)           (MAP)!
! EXTERNAL NAMES .....: WSHELPP (PROGRAM)           WSHELPM (MAP)!
! TRANSACTION CODE.....: * WSHP           !
!
! EXPLICIT KEYWORDS...:           !
! UPDATED BY.....: XXXXXX           ON : 06/12/2001 AT: 17:09:31           LIB : ACC           !
! SESSION NUMBER.....: 0378           LIBRARY.....: ACC           LOCK.....:           !
! *** END ***           !
! O: C1 CH: 0 ZRHELP           ACTION:           !
+-----+

```

```

+-----+
! DOCUMENTATION SCREEN          FLIGHT RESERVATION MENU          PAGE:01/01!
!
!
! INTRODUCTION
!
! With this transaction you can:
!
! * CONSULT FLIGHT CHARACTERISTICS
! * CONSULT THE EXISTING FLIGHTS FOR A SPECIFIC
!   DESTINATION
! * RESERVE PLACES FOR A SPECIFIC FLIGHT
! The Flight management can perform FLIGHT UPDATING.
!
!
! GENERAL MENU
!
! Withé this screen you access screens of   é
! the transaction,è after havinég indicated a password.
!
! CHOICE.....: S   E: END   - P: PREVIOUS - N: NEXT - D: TOP - L: LAST
! PAGE - LINE.:   NUMBER OF PAGES NEXT OR PREVIOUS - LINE NUMBER
!
+-----+

```

Description of the Error Message File

The System generates an error message file. The records generated for this file are described below.

The physical characteristics of this file are:

```

Organization: indexed or relational
Access key   : position 1
              29 characters
Records      : 100 characters

```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	3		LIBRARY CODE
2	1		ENTITY TYPE
			Used to specify the type of entity.
		H	. For screens (Pacbench C/S error messages).
		I	. Record reserved for internal use by Pacbench C/S. It is used by the HELP Function to find the code of the Data Element corresponding to the line / Column position of the cursor.
3	1		NON-USED FIELD
4	6		ENTITY CODE
5	4		SEGMENT CODE/LOGICAL VIEW
6	6		DATA ELEMENT CODE
7	1		CODE TYPE
		BLANK D S U	Allows you to indicate the type of coding: Title Comments Standard error User error (dialogue-related error)
8	4		ERROR NUMBER
			There are three ways to build the error number:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			1. AUTOMATIC ERROR MESSAGE, or EXPLICIT MANUAL ERROR MESSAGE 'ON A DATA ELEMENT',
		1	. associated to code type 'D', indicates a value of the Data Element
		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'
		0	Documentation will be generated before the Data Element description lines.
		2 4 5 6 7 8	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.
			4. AUTOMATIC SEGMENT OR ERROR NUMBER MESSAGES
		DUPL	Invalid creation
		NFND	Invalid modification/deletion
		END	End of list
		ABSC	Absence of record
9	3		LINE NUMBER
			This number is managed by the system.
		000	Error messages
		001-999	Documentary messages
			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.
10	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.
11	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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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.
12	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 8. Access Mode

Screen : 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
LN0tt0eeeee	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
0aabbbbATccccc	Text assigned to Screen 'aabbbb' (starting with text 'ccccc').	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
0aabbbbCSdddd	Call of segments of screen 'aabbbb' (starting with category indicator 'c' and segment dddd). NOTE: "c" is equivalent to " " for the screen-top category.	YES

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

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

Screen Generation and/or Printing

From the Generation and Print Commands (GP) screen, you can request either screen descriptions, or screen description AND generation of the components that can be generated with the Pacbench C/S. Depending on which OPERATION code is entered, the requested output is:

C/S SCREEN DESCRIPTION: DGC

- OPTION C1 - description of screen layout
 - description of Segments
 - description in structured code
- OPTION C2 - description of screen layout
 - description of assigned texts
- OPTION C3 - description of screen layout
 - description of assigned texts
 - description of Segments
 - description in structured code

C/S SCREEN DESCRIPTION AND GENERATION: GGC

- OPTION C1 - description of Screen layout
 - description of Segments
 - description in structured code
 - Screen generation
- OPTION C2 - description of screen layout
- or C3 - description of assigned texts
 - description of Segments
 - description in structured code
 - Screen generation

BUSINESS COMPONENT/FOLDER/FOLDER VIEW DESCRIPTION: DGS

- OPTION C1 - description of Segments
 - description in structured code
- OPTION C2 - description of assigned texts
- OPTION C3 - description of Segments
 - description in structured code
 - description of assigned texts

DESCRIPTION-GENERATION BUSINESS C./FOLDER/FOLDER VIEW: GGS

- OPTION C1 - description of Segments
 - description in structured code
 - Screen generation
- OPTION C2 - description of Segments
- or C3 - description of assigned texts
 - description in structured code
 - Screen generation
- OPTION C4 - description of Segments and generation of the working areas necessary to the call of a server component.

EXTRACTION PRIOR TO PROXY OBJECT GENERATION: GVC

- OPTION C1 - description of Proxy object

>>>> For more information on the GVC command, refer to the 'Pacbench C/S - Business Logic' manual.



Part Number: DDOA000301A - 6048

Printed in U.S.A.

(1P) P/N: DDOA000301A - 6048

