

AUDIN

Composants & systèmes d'automatisme
7 bis rue de Tinquex - 51100 Reims - France
Tel. +33(0)326042021 • Fax +33(0)326042820
<http://www.audin.fr> • e-mail info@audin.fr



EasyGen™

User's Manual



EasyGen™
USER'S MANUAL



DATALOGIC S.p.A.
Via Candini 2
40012 - Lippo di Calderara di Reno
Bologna - Italy

EasyGen™ - User's Manual

Vers.: 1.00 and later
Ed.: 04/01
Code: 820042310

ALL RIGHTS RESERVED

Datalogic reserves the right to make modifications and improvements without prior notification. Product names mentioned herein are for identification purposes only and may be trademarks and or registered trademarks of their respective companies.

© - 2001 Datalogic S.p.A.

DATALOGIC S.p.A. Software License Agreement

This legal document is an agreement between you, the end user and DATALOGIC S.p.A. BY INSTALLING THE SOFTWARE, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, which includes the SOFTWARE LICENSE, LIMITED WARRANTY and ACKNOWLEDGMENT.

GRANT OF LICENSE. DATALOGIC grants to you the right to use one copy of the enclosed DATALOGIC S.p.A., program (the "SOFTWARE") on a single terminal connected to a single computer (i.e.; with a single CPU). You may not network the SOFTWARE or otherwise use it on more than one computer or computer terminal at the same time.

COPYRIGHT. The SOFTWARE is owned by DATALOGIC S.p.A. or its suppliers and is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (i.e.; a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the software.

NON PERMITTED USES: Without the express permission of DATALOGIC S.p.A., you may not:

1. Use the software in a computer service business including rental, networking or time sharing software, nor may you use it for multiple users, or multiple computer system applications in the absence of individual network licenses with DATALOGIC S.p.A.
2. Use, Copy or modify, alter or transfer, electronically or otherwise, the software or documentation except as expressly allowed in this agreement.
3. Translate, reverse engineer, de-assemble, de-compile or create derivative works based on the written materials.
4. Sub-license or lease this program or its documentation.

LIMITED WARRANTY: This software and accompanying written materials are provided "as is" without warranty of any kind. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THIS SOFTWARE IS ASSUMED BY YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE IN USE, YOU (AND NOT DATALOGIC OR ITS AGENTS) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION OF DATA. FURTHER, DATALOGIC DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF, OR THE RESULTS OF THE USE OF THE SOFTWARE, IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE AND YOU RELY ON THE SOFTWARE AND RESULTS AT YOUR OWN RISK.

DATALOGIC warrants to the original licensee that the CD-Rom on which the SOFTWARE is recorded is free from defects in materials and workmanship under normal use and service for a period of ninety (90) days from the date of delivery as evidenced by a copy of your receipt. DATALOGIC's entire liability and your exclusive remedy shall be at DATALOGIC's option, replacement of the CD-Rom which shall be returned to DATALOGIC with a copy of your receipt. If failure of the CD-Rom has resulted from accident, abuse or misapplication of the product, then DATALOGIC shall have no responsibility to replace the product under this warranty.

THE ABOVE IS THE ONLY WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, THAT IS MADE BY DATALOGIC ON THIS DATALOGIC PRODUCT. IN NO EVENT SHALL DATALOGIC OR ITS SUPPLIERS, NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION OR DELIVERY OF THIS PRODUCT, BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE, THE RESULTS OF USE, OR INABILITY TO USE SUCH PRODUCT, EVEN IF DATALOGIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ACKNOWLEDGMENT: You acknowledge that you have read this LICENSE and Limited Warranty, understand them and agree to be bound by their terms and conditions. You also agree that the LICENSE and Limited Warranty are the complete and exclusive statement between the parties and supersedes all other communications between the parties relating to the subject matter of the LICENSE or the Limited Warranty.

CONTENTS

1	GENERAL INFORMATION	1
1.1	Scope of this Manual	1
1.2	Description of the Applications Package	1
1.3	Package Contents	2
1.4	Minimum Configuration Required	2
2	INSTALLATION	3
2.1	Installing EasyGen™	3
2.2	Connecting the Terminal to the Host Computer via RS-232	4
2.3	Loading EasyGen™ Interpreter	6
2.4	Loading a Sample Program	6
3	USE OF EASYGEN™	7
3.1	Introducing EasyGen™	7
3.2	Starting EasyGen™	10
3.3	Limits of EasyGen™	12
3.4	Error Reporting	13
4	MENU DESCRIPTION	14
4.1	Description of the Main Menu	14
4.2	File Menu	15
4.2.1	New	15
4.2.2	Open	16
4.2.3	Save	19
4.2.4	Save as	19
4.2.5	Print Setup	19
4.2.6	Print	20
4.2.7	Send	20
4.2.8	Export	21
4.2.9	Exit	22
4.3	Edit Menu	23
4.3.1	Search	24
4.4	Insert Menu	25
4.5	Run Menu	26
4.5.1	Compile	26
4.5.2	Start	27
4.5.3	Stop	28
4.5.4	Step	28
4.6	Tools Menu	29
4.6.1	Editor Tables	30
4.6.2	Editor Variables	35
4.6.3	Options	36

4.7	Utility Menu	38
4.8	Window Menu	39
5	INSERT MENU FUNCTIONS	40
5.1	Remark Function.....	40
5.2	Beep Function.....	41
5.3	Clear Display Function.....	42
5.4	Input Barcode Function	43
5.5	Input Text Function	47
5.6	Input Number Function	49
5.7	Input Date Function.....	51
5.8	Input Time Function	52
5.9	Menu Function	53
5.10	Loop Function	54
5.11	Comparison Function.....	55
5.12	Go to Function	56
5.13	Mathematical Operation Function	57
5.14	Find Function	60
6	EASYGEN™ INTERPRETER PROGRAM.....	62
6.1	Interpreter Program Use	62
6.2	Use of the Formula Terminal.....	62
6.2.1	Description of Keys	63
6.2.2	Select Table	65
6.2.3	Data Display.....	66
6.2.4	Data Deletion	67
6.2.5	Data Search	68
6.2.6	Error Messages.....	71
6.3	Deletion of the Application Program.....	74
6.4	Deletion of the Interpreter program.....	75
A	PROGRAMMING EXAMPLES	77
A.1	Example 1: Code+Quantity	78
A.2	Example 2: EAN128 3 Fields	82
A.3	Example 3: Assisted Sales.....	86
A.4	Example 4: Listed Picking	91
A.5	Example 5: Inventory	94
B	DOSDL	97
B.1.	Loading Files Using DOSDL.EXE	97
B1.1.	DOSDL.EXE - Interpreter Program	98
B1.2.	DOSDL.EXE - Application program	100
B1.3.	DOSDL.EXE - Tables	102
B.2.	Alternative File Loading Method.....	104

C COMMUNICATION PROTOCOL..... 106

C.1. Communications frame 106

C.2 Data Transmission 109

C2.1 RS-232 and RS-485 Interfaces 109

C.2.2 Eavesdrop Connection Interface..... 109

C.3 Host Computer Commands..... 110

C.4 Configuring the Terminal (Serial Command “S”) 115

Formula[®]

1 GENERAL INFORMATION

1.1 SCOPE OF THIS MANUAL

This manual was created by Datalogic S.p.A. and accompanies the EasyGen[™], Easy Application Generator Package, CD-Rom for Windows 95/98/2000/ME and NT.

The manual describes the methods for installation and use of both EasyGen[™] and its relative interpreter program. It is also designed for those who are familiar with the Microsoft Windows 95/98/2000/ME/NT operating environment and basic programming techniques.

1.2 DESCRIPTION OF THE APPLICATIONS PACKAGE

EasyGen[™] is an applications package for Windows 95/98/2000/ME and Windows NT environments, designed to simplify the creation of personalized software for Formula batch terminals. It facilitates the creation of programs so that even users who are not programming experts can create software which can be transferred to Formula batch terminals.

EasyGen[™] permits the user to define: the sequence of functions which the Formula batch terminal should execute, the interactions with the user, and the use of data or support tables. It loads the interpreter program, the application program, and the associated tables onto the Formula batch terminal using the relative cradle connected to the PC via an RS-232 port.

Currently EasyGen[™] functions on Formula batch terminals F734, F732, F725 and F660.

1.3 PACKAGE CONTENTS

The EasyGen™ package includes:

- 1 CD-Rom containing the following programs:
 - EasyGen™ with example programs
 - EasyGen™ Interpreter
 - SysTools™2001
 - OLE for Formula
 - DOSDL program
- The following manuals:
 - EasyGen™ (this manual)
 - SysTools™2001
 - OLE for Formula

1.4 MINIMUM CONFIGURATION REQUIRED

In order to install and use the EasyGen™ program, it is necessary to have at least the following:

- PC with a Pentium processor;
- one of the following operating systems:
 - Windows 95 - SR2 or later
 - Windows 98
 - Windows 2000
 - Windows ME
 - Windows NT + Service Pack 4
- Internet Explorer 4.01 SP2 (installed in the Active Desktop);
- *HTML Help Viewer 1.32 (available in the EasyGen™ installation directory);
- free serial port;
- 16 MB memory for Windows 95/98, 32 MB for Windows ME/2000/NT;
- hard disk with at least 30 MB free space to run;
- VGA video card or better.

* Advised only for Windows NT 4.0 or Windows 95.

Formula[®]

2 INSTALLATION

2.1 INSTALLING EASYGEN™

Once the CD is inserted, launch the Windows software application EasyGen™ for WIN 95/98/2000/ME/NT by pressing the Setup button. Through this procedure the following programs will be installed:

- **EASYGEN.EXE** Permits the creation and modification of personalized application programs and their loading onto a Formula batch terminal.
- **DOSDL.EXE** Allows you to load the Interpreter Program, the application program (which was created with EasyGen™), and the tables associated with it onto a Formula batch terminal with a PC running DOS.
- **EIXXX_YYY.HEX** Permits the interpretation by Formula terminals of operations defined with the application program EasyGen™. The characters xxx refer to the terminal name, while yyy indicate the software version.
- **SYSTOOLS™
2001** Permits the connection of the computer to the terminal via a cradle, for configuration and the creation of archives containing collected data.

- OLE for Formula is for very skilled users only. It can be used to load applications and collect data from the Formula batch terminals from within your own application, using any Development System for Windows that supports the “Microsoft OLE” Technology.

2.2 CONNECTING THE TERMINAL TO THE HOST COMPUTER VIA RS-232

It is necessary to make the proper RS-232 connection for your PC to the Formula batch terminal.

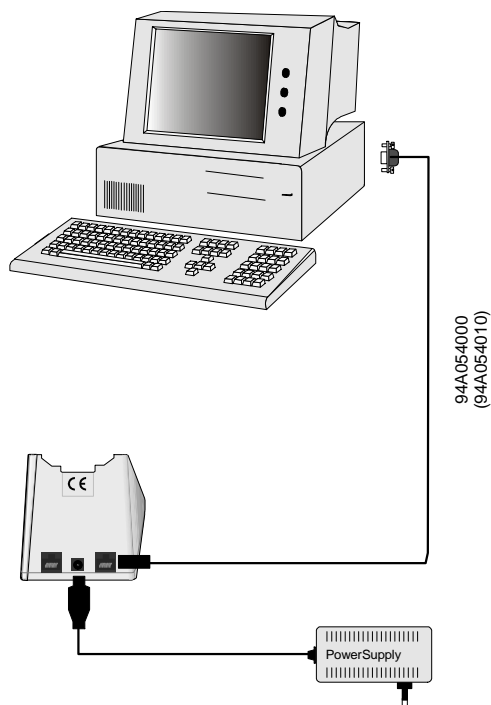


CAUTION

Before proceeding to this phase, make sure that both the computer and the terminal are turned off.

To connect the Formula cradle to the host computer, proceed as indicated below:

1. connect the serial port of the host computer to cable 94A054000 for 9-pin connections or to cable 94A054010 for 25-pin connections;
2. connect the other end of the same cable (RJ connector) to the RS-232 port of the cradle;
3. insert the power-supply plug into the jack on the base of the cradle;
4. attach the power supply to a power outlet;
5. turn on the cradle and the computer;
6. put the Formula terminal into the cradle.



2.3 LOADING EASYGEN™ INTERPRETER

Load the EasyGen™ Interpreter Program EIXXX_YYY.HEX onto the Formula terminal with the EasyGen™ command "INITIATE" (see par. 4.7). You can also load the EasyGen™ Interpreter Program with a DOS system using the DOSDL.EXE program (see Appendix B). Before proceeding with this operation verify if another interpreter program has been previously loaded. In this case it is required to delete the program by using the shielded reset key (see par. 6.2.1 and par. 6.4 for details).

2.4 LOADING A SAMPLE PROGRAM

Select an EasyGen™ sample application, the simplest one is "EXAMPLE1.APP" found in the "Installation Directory \ Examples \ *Your Language* \ EXAMPLE 1 (Code+Quantity)", and load it onto the Formula terminal using the "SEND" command (see par. 4.2.7 for details).

You should be able to run the loaded application on the Formula terminal. The example application "EXAMPLE1.APP" helps you to build a program which stores collected barcodes, date and time in a table.

To collect data from the sample application, use the SysTools™2001 program. Follow the SysTools™2001 User's Manual for details on how to install and use this program to collect data from Formula terminals.

Formula®

3 USE OF EASYGEN™

3.1 INTRODUCING EASYGEN™

EasyGen™ is the fastest and easiest way to create applications for many Datalogic hand held terminals. Whether you are an experienced professional or brand new to hand held scanner programming, EasyGen™ provides you with a tool to simplify rapid application development.

Rather than writing numerous lines of code to create your application running on the terminal, you add simple functions into a list on the PC screen.

The functions, properly described in chapter 5, give the EasyGen™ user access to the main features of the terminal.

A full list of available functions is:

- Remark to write text on the display;
- Beep to make a sound;
- Clear display
- Input barcode define the scanner barcode reading;
- Input text to key in or read with the scanner alphanumeric text;
- Input number to key in or read with the scanner a number;
- Input date to read the date from the real time clock of the terminal;
- Input time to read the time from the real time clock of the terminal;
- Menu to display different choices and allow the user to select one;
- Loop to go to a function and update the tables/indexes (see paragraph "Editor tables" on page 30);
- Comparison to make a comparison between numbers or text;
- Go to to go to a function;
- Mathematical operation to make a basic mathematical operation;
- Find to find text or a number inside a table (see paragraph "Editor tables" on page 30);

If your operation must store data, either previously prepared tables, or an output file for uploading to the PC, you must define several different tables, each one composed of a fixed number of records in which you have different fields (see paragraph 4.6.1).

To develop a new program with EasyGen™ we suggest that you proceed with the following steps:

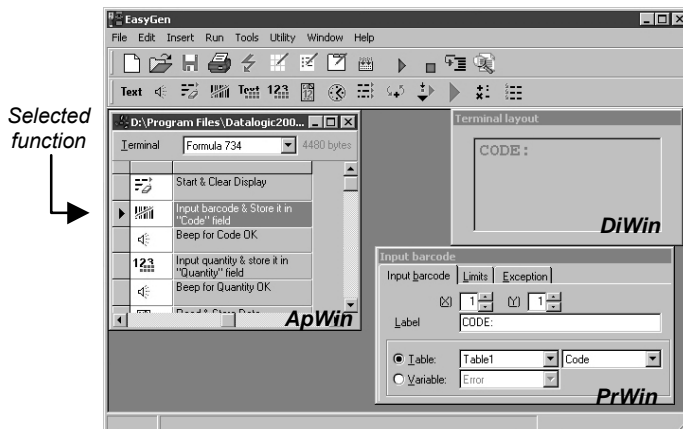
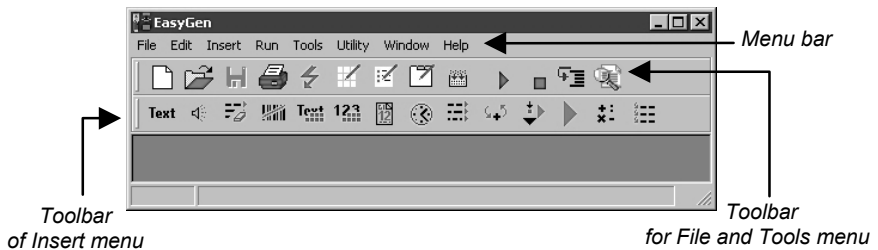
1. perform an analysis of the application and build a flow chart using the available functions;
2. open a new program [File menu - New] see par. 4.2.1;
3. choose the type of Formula terminal to be used;
4. define the tables and fields [Tools menu - Editor Tables] see par. 4.6.1;
5. define names for variables [Tools menu - Editor Variables] see par. 4.6.2;
6. proceed with the insertion of all the functions that must be executed by the terminal as defined by your flow chart [Insert menu] see par. 4.4;
7. save the program [File menu - Save] see par. 4.2.3;
8. set the Formula terminal with the serial parameters that must be used by your application [Tools menu - Options] see par. 4.6.3;
9. send the interpreter program EIXXX_YYY.HEX to the Formula terminal (this "Initialization" operation is only performed the first time that a program must be loaded) [Utility menu - Initiate] see par. 4.7;
10. load the actual application program on the Formula terminal [File menu - Send] see par. 4.2.7.

3.2 STARTING EASYGEN™

To start EasyGen™ proceed as follow:

1. Click "Start" on the task bar, Select "Programs", "Datalogic", "EasyGen" and then "EasyGen™" (this path corresponds to the default installation);
- or -
Use the Windows Explorer to find the "EasyGen™" executable file (Setup.EXE);
2. Double click the EasyGen™ icon.

When you first start EasyGen™, you see the interface of the development environment, as shown in the following figures:



The development environment interface of EasyGen™ is similar to other programs for Windows 95/98/2000/ME/NT.

The main elements are:

- Menu bar displays the commands you use to work with EasyGen™;
- Tool bar Button Provides quick access to commonly used commands in the programming environment; The icons on the first line correspond to the selections of the File and Tools menu, while the icons of the second line correspond to those of the Insert menu.
- Application Window (ApWin) Lists, from top to bottom, the functions of your application as they are executed by the terminal when loaded;
- Display Window (DiWin) Emulates the display of the selected terminal as it appears after the complete execution of the selected function;
- Property Window (PrWin) Lists the property settings for the selected function.

Chapter 5 will give a more detailed description of these windows.

3.3 LIMITS OF EASYGEN™

Message/Comment:	This is a sequence of a maximum of 32 alphanumeric characters;
Number:	<p>is formed by a maximum of 7 integers and two decimal numbers in the format</p> $\pm\text{XXXXXXXX.YY}$ <p>The programmer is responsible to ensure that all numerical operations do not exceed this limit;</p>
Variables:	7 temporary variables of a maximum of 32 alphanumeric characters are available. The first one is reserved as an "Error Variable" the other 6 can be named (see par. 4.6.2).
Tables:	It is possible to use simultaneously up to 15 named tables (1..15). In general, table 0 is used to store acquired data. A numeric index variable is associated with each table. Tables are composed of records and each record can have a maximum of 30 fields (see par. 4.6.1). The index variable points to the nth record. It assumes the minimum value of zero, or "first record", and the maximum value of "maximum number of records" minus one.

3.4 ERROR REPORTING

The “Error Variable” reports these possible error conditions:

Error type	Variable value
No error	00
<F1> button has been pressed	1
Division by zero	10
Table limit reached	30
Data code not suitable for destination field	31

It is the responsibility of the programmer to make sure that the inserted function does not generate one of these errors by using the comparison function.

The “Error Variable” must be set to the value “00” at program start and reset after each reporting of an error value.

Formula®

4 MENU DESCRIPTION

4.1 DESCRIPTION OF THE MAIN MENU

In this chapter we will describe the functionality of each item of the menu bar and the corresponding toolbar button available for it.

Instead of using the menu bar, it is possible to make the same actions by clicking the buttons. The Tool bar buttons show the tool-tips when you point at them with the cursor.

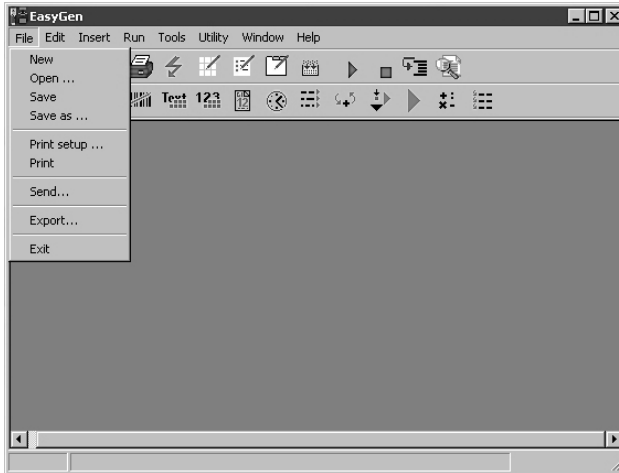
The main menu of the EasyGen™ environment appears at the top of the window when you start the program.

It's composed of the following items:

- File file support, printing, application loading;
- Edit moving and deleting functions;
- Insert inserting functions in the application;
- Run emulation, under windows, of the Formula terminal;
- Tools parameters editing;
- Utility load interpreter on the Formula terminal;
- Window multiple windows setting;
- Help help.

4.2 FILE MENU

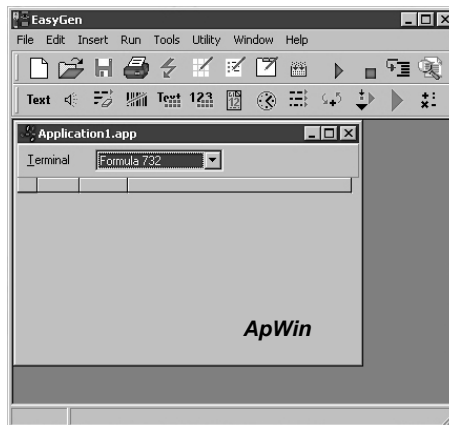
Manage application file, printing and loading to the Formula terminal.



4.2.1 New



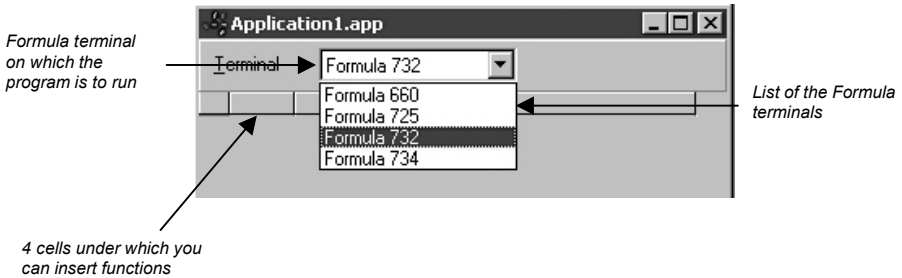
Choosing this command allows you to create a new application program. An Empty Application Window (ApWin) appears on the screen:



The upper part of the ApWin shows the type of Formula terminal the program is to run on. With the down arrow, it is possible to choose the terminal type and EasyGen™ adapts the base program to the model of the Formula terminal chosen.

Below the Formula terminal name there is row with 4 cells under which you can insert functions using the Insert command or function tool bar button.

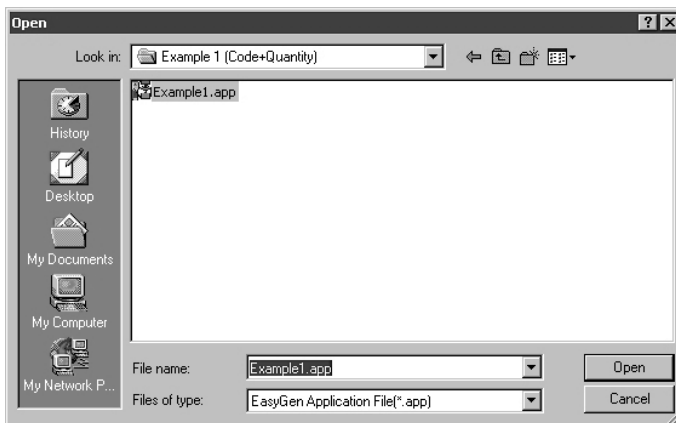
The description of the 4 cells is given under the Open command (see par. 4.2.2).



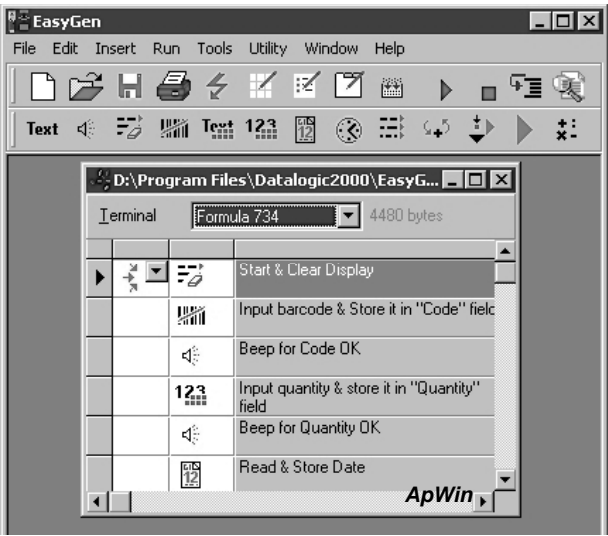
4.2.2 Open



With this command it is possible to open a previously created application program. The Windows 95/98/2000/ME/NT File selection dialogue window appears on the screen.

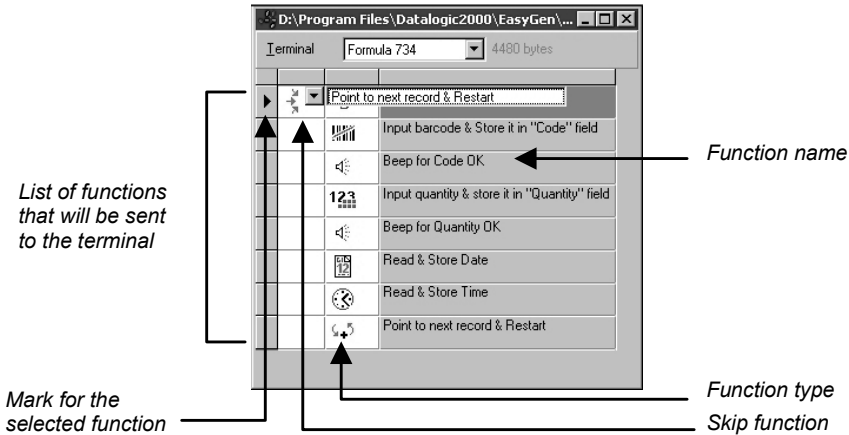


The ApWin that appears contains the application program - that is, the sequence of functions that must be sent to the Formula terminal:

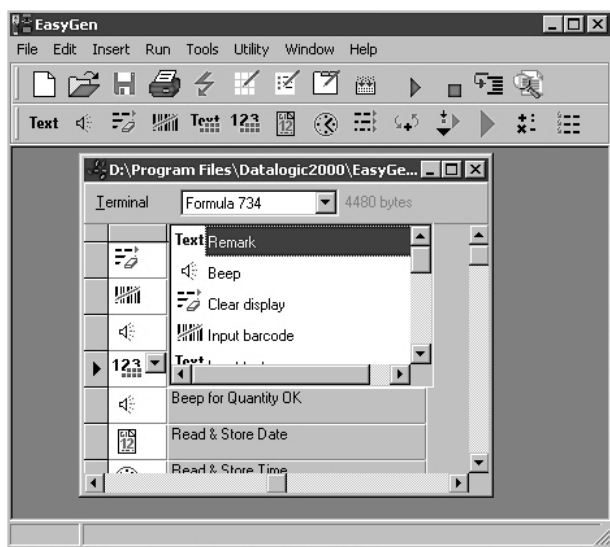


In the ApWin there is the list of functions that are performed by the application in the Formula terminal.

The four columns, as seen in the figure below are used for (from left to right):



- **Mark** cell, with an arrow, points to the selected function in editing mode;
- **Skip function** cell, with multiple red arrows to one point, shows that this function is connected by a jump or a loop (see par. 5.10). The little arrow button on the cell can be used, if selected with the pointing device, to see the name of the calling function.
- **Function name** cell, shows the unique name assigned to the function. The name is used to connect jump or loop functions to this one. By selecting this box it is possible to change the name of a function directly within the window; the name can be of any length and identifies the function. The system suggests a default name with progressive numbers automatically.
- **Function type** cell, shows the symbol of the function used. Pointing to the cell causes a little arrow to appear in the right upper corner. By clicking on it, it's possible to change the function type by selecting from the list that appears.



The function position can be changed within the ApWin through the Drag and Drop procedure. If the CTRL key is pressed simultaneously, the dragged function will then be copied.

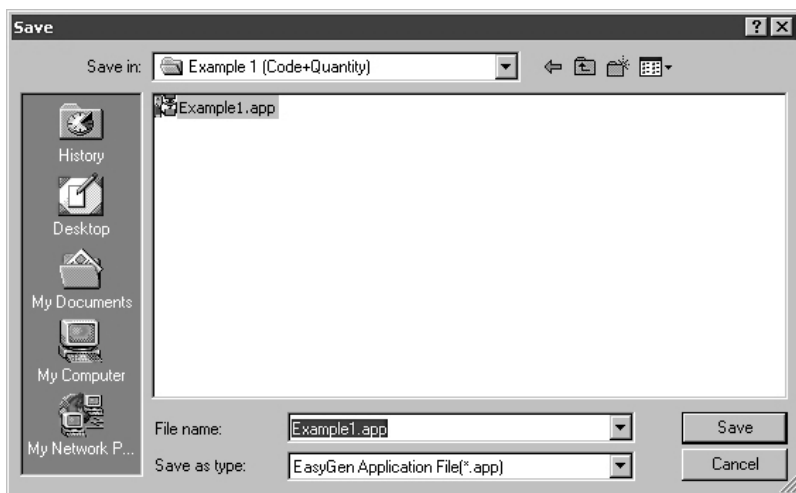
4.2.3 Save



This choice allows you to save an application program which you have created. This command saves immediately the currently open application. Note that it will not prompt for a name. If the open program is not already named, EasyGen™ will assign one of the format Applicationx.APP, where x is a progressive number. A file with the same name and PWT extension (Applicationx.PWT) will be automatically saved. This file contains the name of Formula terminal and is useful for some terminal-dependent instructions. If this file does not exist “Formula 734” will be the default terminal.

4.2.4 Save as ...

This command can be useful when you wish to modify a program and save it to disk with a different name, without modifying the original.



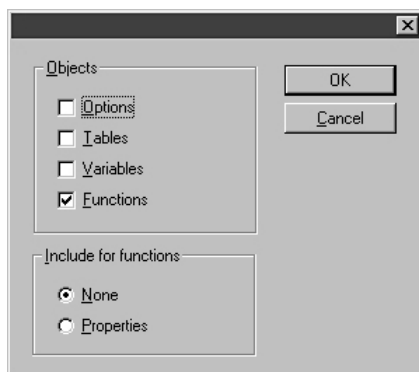
4.2.5 Print Setup

This command allows you to select a printer and to specify print options, page size, orientation, etc... in a standard printer setup dialog of Windows 95/98/2000/ME or NT.

4.2.6 Print



This command allows you to print the current application program. The following dialogue box appears on the screen:



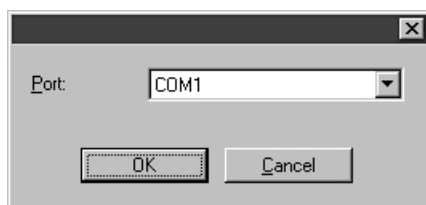
It is possible to choose which portions of the application program to print:

Options/Tables/Variables: The information defined with the Tools menu is printed;
Functions: The program is printed with the functions defined by the Insert menu.
It is also possible to select printing of the function properties desired.

4.2.7 Send



With this command it is possible to transmit the application program and the tables to the Formula terminal. The following dialogue box appears on the screen:



It is necessary to indicate which port the Formula terminal is connected to.

At the end of the transmission, the date and time of the terminal will be the current date and time you have in the PC.

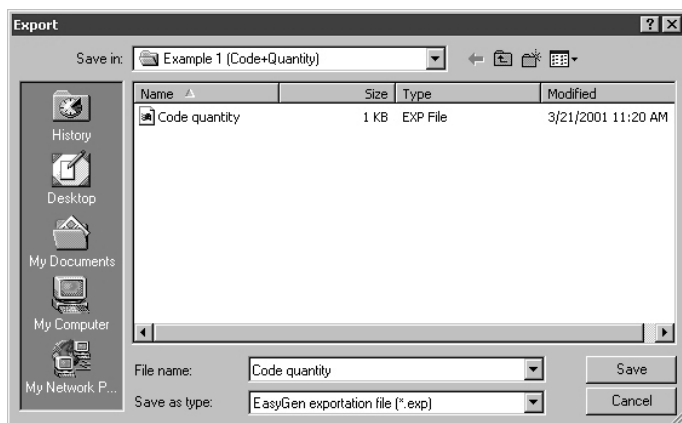
This operation is also possible in a DOS environment using the DOSDL.EXE command (See Appendix B).

**NOTE**

Before sending a new application program to the terminal, it is necessary to delete the one currently in use. See par. 6.3 "Deletion of the application program".

4.2.8 Export

This command is used to save your application specific files to be loaded by an operating system other than Windows 95/98/2000/ME or NT. The Program DOSDL.EXE under DOS can be used for example.



The "Export" command produces files with the same name as your application, appl, and with different extensions for different sequential loading: <appl.EXP>, <appl.CMD> and <appl.CFG>.

**NOTE**

DOSDL recognizes application filenames having a maximum of 8 characters.

It is advised to select filenames that respect this limit before Exporting them from EasyGen™.

**CAUTION**

The application program and the tables assigned to it must be loaded onto the Formula terminal in compliance with the communication frame described in Appendix C.

This command also produces the table files (.INP) with the name of the tables that are defined in your application program.

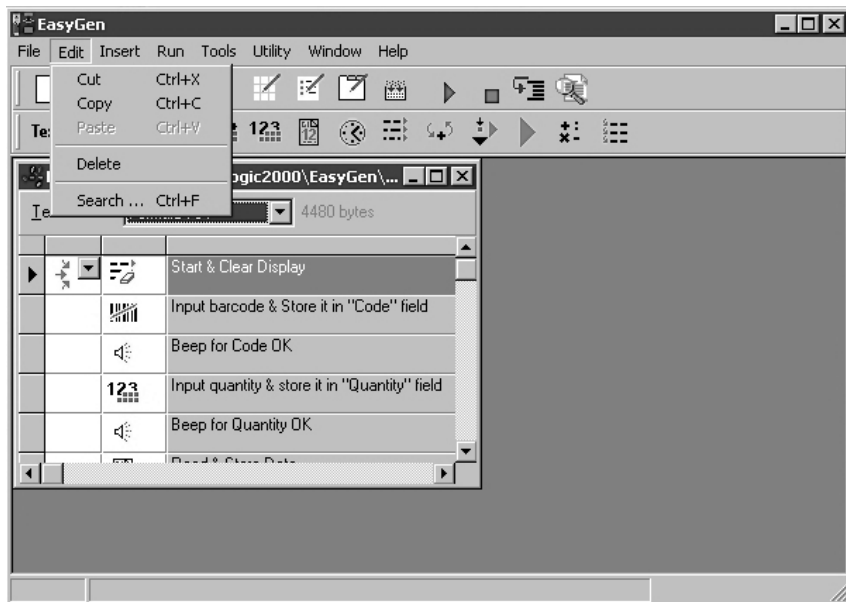
Each table.INP file contains a description of the structure of the table. It is the responsibility of the programmer to have prepared an ASCII table.DAT file, as defined in par. 4.6.1, in cases where the table should be filled in with default or constant data.

To load files with other methods see Appendix B.

4.2.9 Exit

Select this command to exit EasyGen™.

4.3 EDIT MENU



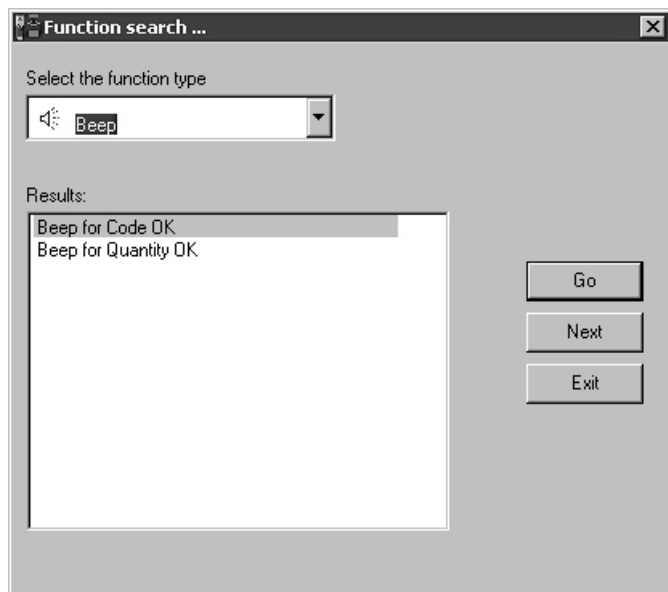
The commands of this menu allow you to copy, insert, delete and search for program functions.

The command Paste inserts a previously copied function, below the current line. If more than one program is opened, it is possible to copy or move a group of functions from one program to another.

4.3.1 Search



This command allows the user to search for a function inserted in the program. In this case the following window appears:



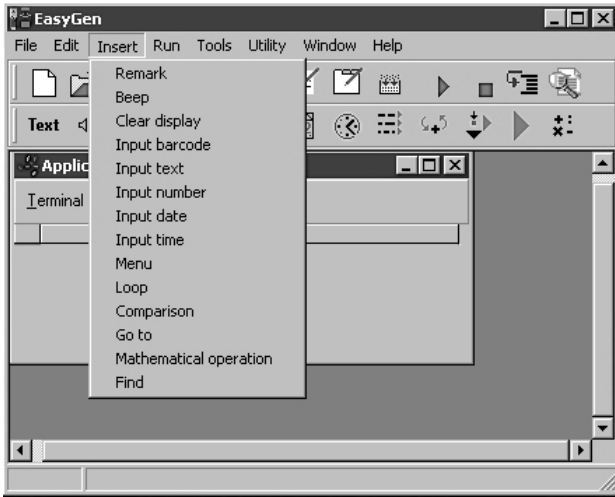
Once a function name has been chosen from the pull-down list, in the "Results" area you will be presented with a list of the occurrences of that function in the program you are working with.

By selecting a specific function in the "Results" area and pressing the "Go" key, its position is immediately located and pointed to by the arrow in the program ApWin. The "Function Search" window will be closed. The same procedure can be obtained by double-clicking on the function name from the "Results" area.















The "Next" key allows scrolling the "Results" area list and therefore locating the occurrences of program functions (scrolling). The "Function Search" window will not be closed.

4.4 INSERT MENU

This menu shows the list of all the available functions for creating the program.



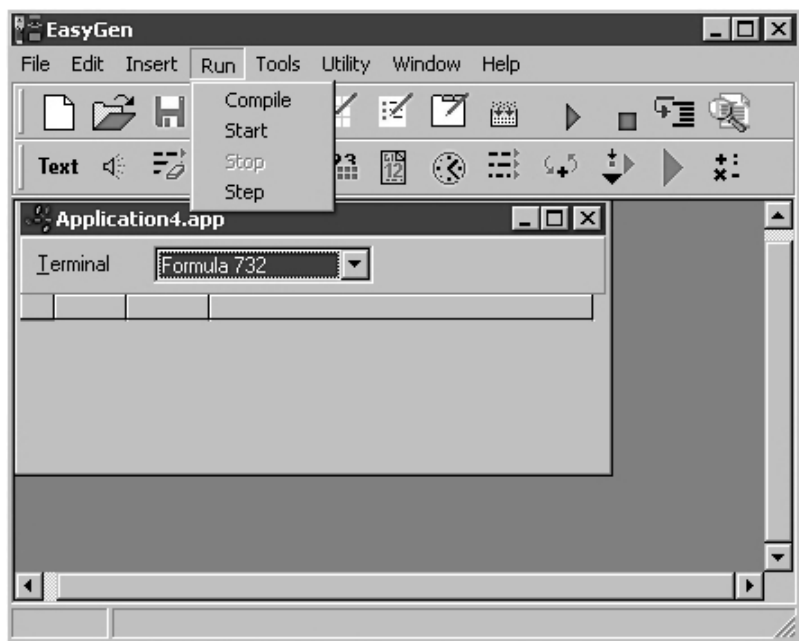
Each function can be inserted into the program by selecting the choice from the menu, or by selecting its corresponding icon.

	Remark		Input time
	Beep		Menu
	Clear display		Loop
	Input barcode		Comparison
	Input text		Go to function
	Input number		Mathematical operation
	Input date		Find

A full description of these functions is given in chapter 5.

4.5 RUN MENU

This function will simulate the execution of the application program.



4.5.1 Compile

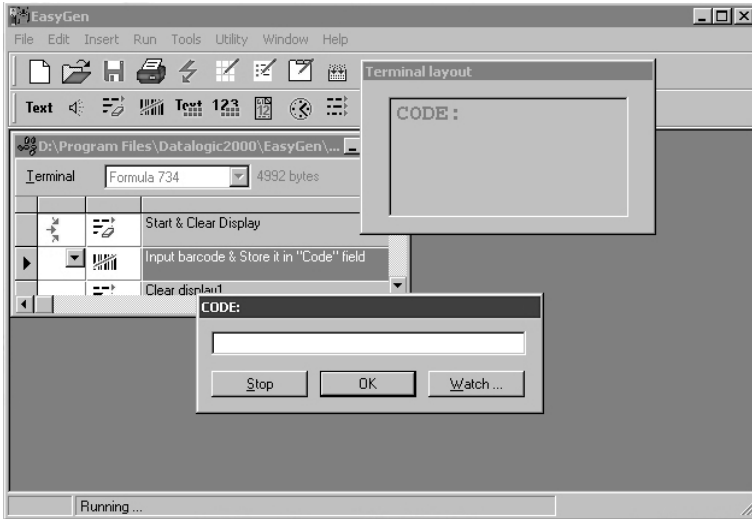


This command allows checking the correct syntax of each function. The compilation is followed by a message. If an error has occurred, the message reports its type, while the arrow in the program ApWin will signal the cell containing the wrong function.

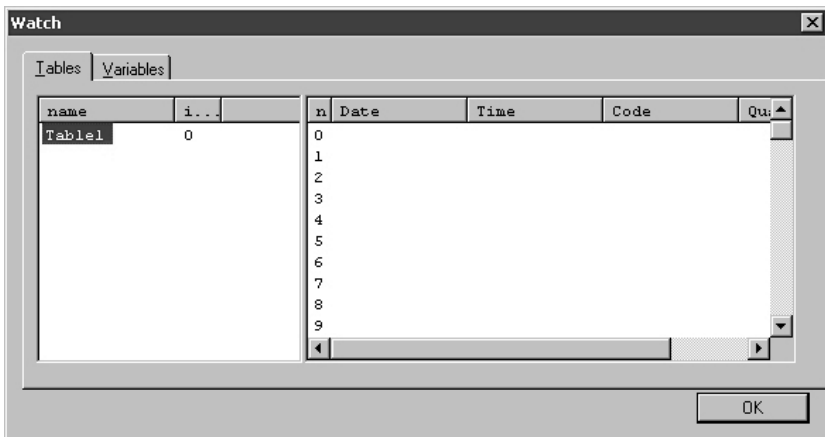
4.5.2 Start

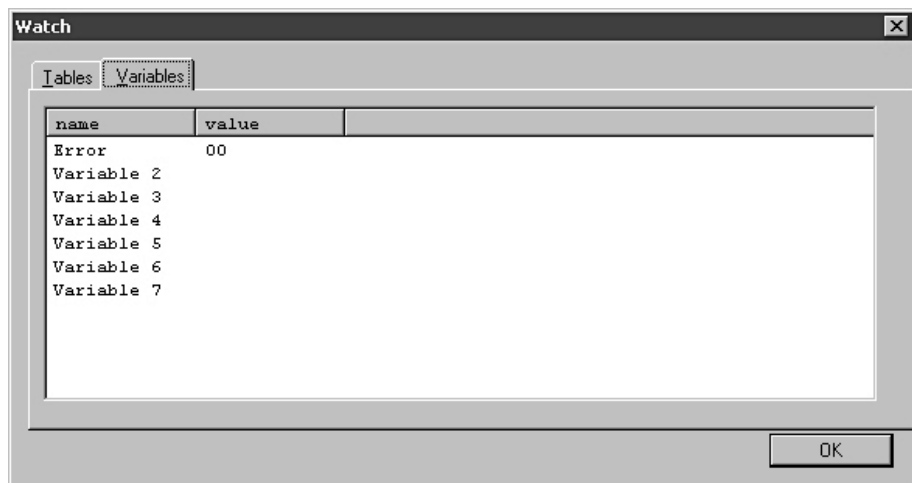


This function allows the user to start the simulation of the application program which runs until there is an acquisition function. In this case a window appears in which you can insert the requested information.



In this operation, if you press the <Watch> button, you can see the tables and variables contents.





4.5.3 Stop



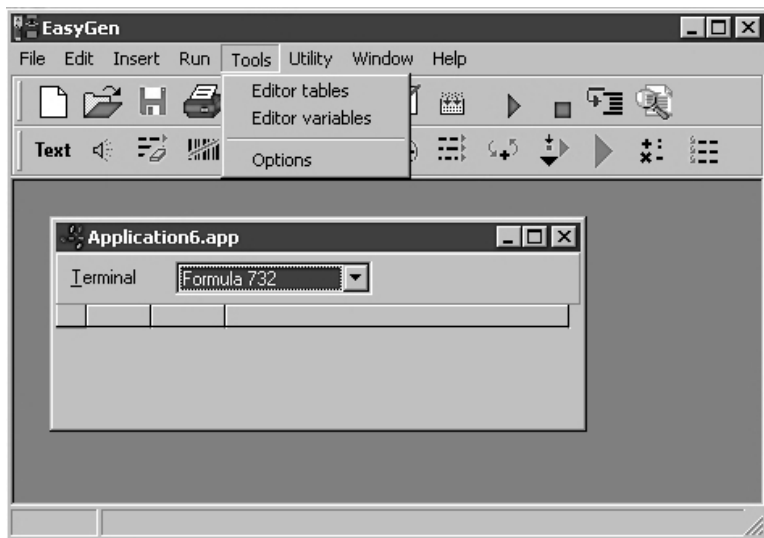
This function stops the simulation of the application program.

4.5.4 Step



This function will force the simulation to go step by step.

4.6 TOOLS MENU



When a EasyGen™ program is open, the Tools menu is activated in order to configure tables, variables, and the terminal.

4.6.1 Editor Tables



Choosing this command causes a window of the following type to be displayed:

The screenshot shows a dialog box titled "Table1". It has two tabs: "General" and "Fields". The "General" tab is active. Inside the dialog, there are several input fields and buttons. The "Number" field contains the value "1" and has left and right arrow buttons. To its right are "Add table" and "Remove table" buttons. Below "Number" is the "Name" field containing "Table1". Below "Name" is the "N. max" field containing "100" followed by the text "records". Below "N. max" is the "EAN128" checkbox, which is currently unchecked. At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

The programmer may define up to 15 General type tables which may be used for reading or writing.

Each table may be associated with a .DAT file which has the same name as the table and which contains its data.

The format of the .DAT file must be compliant with that of the table defined with EasyGen™.

Each field must be divided by the character "|" (pipe) and each record must be positioned on a single line.

EXAMPLE

Suppose that we have defined with EasyGen™ a table "Fruits" consisting of two fields:

Fruit : 10 characters

Code : 5 characters.

The file with extension .DAT is called "FRUITS.DAT" and contains the data:

Pearxxxxxx |00123 |

Orangexxxx |00257 |

Melonxxxxx |00378 |

When the program created with EasyGen™ is sent to a Formula terminal, if files with the extension .DAT (having the same names as tables used within the program itself) are found within the same directory where the program is stored, EasyGen™ loads these data files onto the terminal as well.

**CAUTION**

*Take care when loading tables with external data file (.DAT).
Tables are always loaded starting from the first record, up to the first empty record.
If a table is written in a random order, it is possible that empty records exist between the first and the last records. In this case there will certainly be a loss of data.*

Each table is distinguished by:

☐ **Number**

The number is a progressive number which uniquely identifies the table and corresponds to the order in which it is sent to the Formula terminal.

☐ **Name**

The name of the table is an unique alphanumeric field of maximum 8 characters.

☐ **N. max.**

Maximum number of records which can be read into this table.

Before inserting the number of records which comprise the table, it is necessary to verify the quantity of memory available on the Formula terminal. The maximum number of records allowed depends on the available memory on the Formula Terminal.

☐ **EAN 128**

By checking the EAN 128 box, a special table is defined for EAN 128 barcode reading. The table is automatically assigned the name EAN128tb. This table must have a number of fields greater than or equal to the maximum number of fields to be read. For more details see par. 5.4.

Add table

Add a table

Remove table

Delete a table.



It is not possible to remove tables which have already been called by one or more program functions.

ATTENTION

There is an index for each table which identifies the record where the variable or the result of an acquisition will be stored. The index may be repositioned by the programmer to point to any record in the table.

When the tables are loaded into the Formula terminal, the indexes are always positioned at the first record.

The Fields option allows the definition of the fields making up each record.

Table3

General Fields

Number: 2

Name: Field2

Type: Normal

Normal field

Alignment: Left Size: 32

Fill: Blank Check character from: to z

A record may be a maximum of 256 characters and may contain up to 30 fields. Variable-length records may not be defined.

Each field is distinguished by:

❑ **Number**

The number is a progressive number which uniquely identifies a field and corresponds to the order in which it is sent to the Formula terminal.

❑ **Name**

The name of the field is an alphanumeric field with a maximum length of 10 characters.

❑ **Type**

The field type may be selected from:

- Normal (for example code, price, location, ...)
- Time (internal to the terminal) contains the time in the format hh:mm:ss. Any time a record is updated, the time will be automatically updated as well.
- Date (internal to the terminal) contains the date in the format dd/mm/yy. Any time a record is updated, the date will be automatically updated as well.

For each field of type Normal it is possible to define:

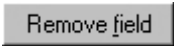
- **Alignment**
It is possible to define the alignment of data within the field, whenever the fill character is not defined as being equal to zero.
- **Fill**
The character to be used to complete acquired data (if the data has a length shorter than the field length).

Space	Data is considered to be alphanumeric.
Zero	Data is considered to be numeric and the alignment is set flush right. This selection assumes relative importance when the data is used (for example, equivalence tests and arithmetic operations).
- **Size**
It is possible to enable control of the length of data to be inserted into the field. Control is done by verifying whether the length is within the defined value. The maximum length allowed is 32 characters. Length control is disabled as the default.

- Check characters from...to
By defining a range of characters, it is possible to enable a control on the type of characters which will be inserted into the field.
By default, character control is between the character "space" and the character "z".
For whole numbers insert the control from "0" to "9". For real numbers insert the control from "." to "9".

A rectangular button with a light gray background and a thin black border. The text "Add field" is centered in a black, sans-serif font.

Add a field

A rectangular button with a light gray background and a thin black border. The text "Remove field" is centered in a black, sans-serif font.

Delete a field



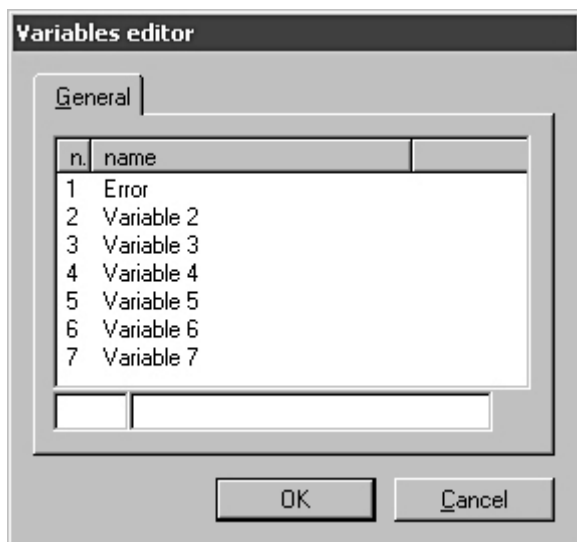
It is not possible to remove fields contained in the tables which have already been called by one or more program functions.

ATTENTION

4.6.2 Editor Variables



Choosing this command causes a window of the following type to be displayed:



The variables are used by the application to store temporary data.

There are 7 temporary variables of 32 characters available.

Each variable is distinguished by a name which enables recognition.

To change the name of a variable it is only necessary to select it, modify the name, and press the <Tab> key to skip to the next variable, or <OK> to exit this window.

4.6.3 Options



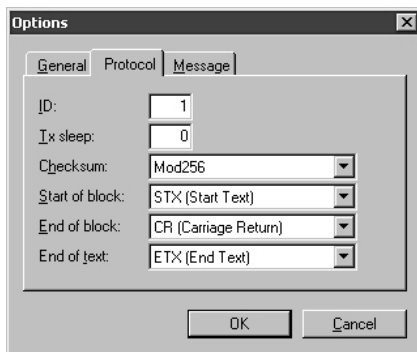
Choosing this command causes a window of the following type to be displayed:

The screenshot shows a dialog box titled "Options" with a close button (X) in the top right corner. It has three tabs: "General", "Protocol", and "Message". The "General" tab is currently selected. Inside the "General" tab, there are five labeled dropdown menus: "Baud rate:" (set to 9600), "Data bits:" (set to 8), "Parity:" (set to Mark), "Communication:" (set to Half duplex), and "Interface:" (set to RS-232). At the bottom of the dialog box, there are two buttons: "OK" and "Cancel".

In this window it is possible to define the communications parameters between the Formula terminal and the host computers.

In the General option the user must enter the speed used by the Formula terminal for communicating with host computer, the number of data bits, the parity, mode, and type of communication (*RS-232, RS-485, and Eavesdrop*).

The Protocol option defines how the records will arrive at the host computer when they are requested to be unloaded (the communication frame is declared in this window).



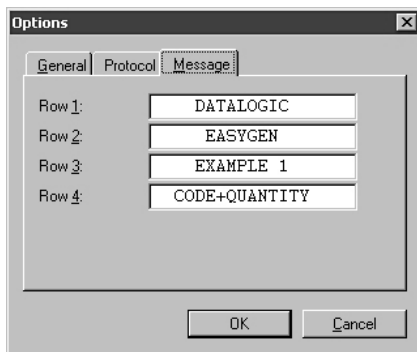
The screenshot shows the 'Options' dialog box with the 'Protocol' tab selected. The 'General' tab is also visible. The 'Message' tab is not selected. The 'ID' field is set to 1. The 'Tx sleep' field is set to 0. The 'Checksum' dropdown is set to 'Mod256'. The 'Start of block' dropdown is set to 'STX (Start Text)'. The 'End of block' dropdown is set to 'CR (Carriage Return)'. The 'End of text' dropdown is set to 'ETX (End Text)'. The 'OK' and 'Cancel' buttons are at the bottom.

The identifier of the Formula terminal can be defined, the transmission delay expressed in milliseconds, the type of message integrity control, and the characters indicating begin block, end block, and end of text.

The suggested values are those required to communicate with the Formula terminal using the Datalogic S.p.A programs such as SysTools™2001.

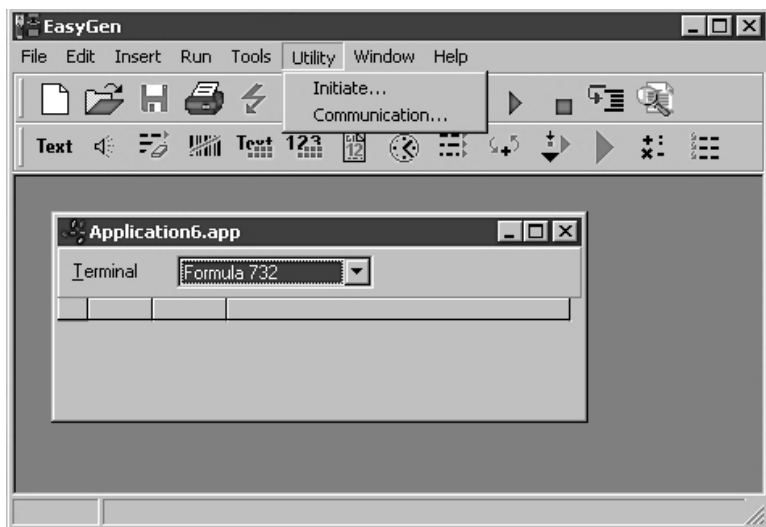
If you use *Eavesdrop* communication (Interface), you have to set: Checksum = *none* and Start of block/End of block = *none*.

The Message option allows the user to define the courtesy message which the Formula terminal displays when it is placed in the cradle.



The screenshot shows the 'Options' dialog box with the 'Message' tab selected. The 'General' and 'Protocol' tabs are also visible. The 'Message' tab contains four rows of text input fields. Row 1 is 'DATALOGIC', Row 2 is 'EASYGEN', Row 3 is 'EXAMPLE 1', and Row 4 is 'CODE+QUANTITY'. The 'OK' and 'Cancel' buttons are at the bottom.

4.7 UTILITY MENU



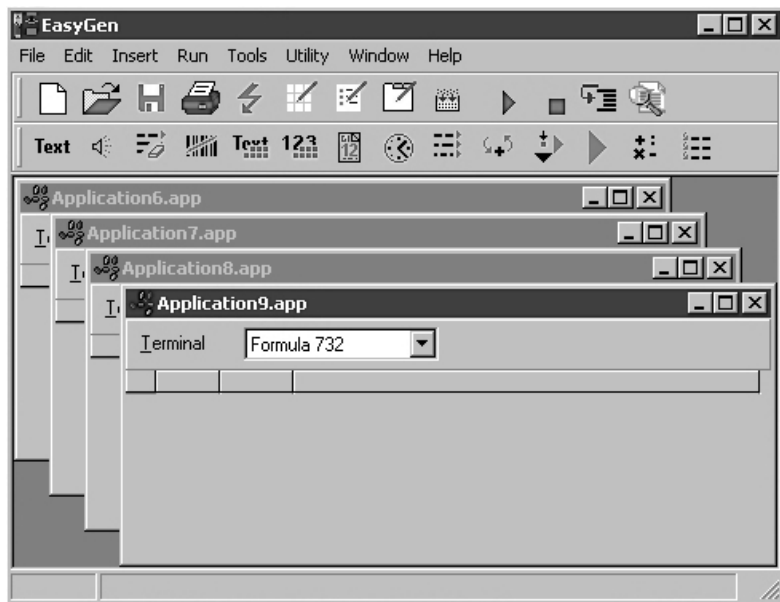
The Utility menu contains two options, "Initiate...", and "Communication...".

Initiate can be used for updating the interpreter program on the Formula terminal. It allows the user to find the file with a .HEX or a .H00 extension in the current directory or in the entire disk.

This operation is also possible using the DOSDL.EXE command in a DOS environment (see Appendix B).

Communication allows selecting the type of data transfer to use towards the terminal.

4.8 WINDOW MENU



The menu option is absolutely Windows standard and allows the user to choose the most convenient display mode when more than one window is opened at the same time:

- Cascading, the windows are displayed with standard dimensions, one on top of the other, but slightly displaced so that each one can be seen.
- Tiled, the windows opened at the moment are automatically scaled and occupy all available space without overlapping.
The windows can be minimized and represented by an icon on the screen.
- Arrange icons, the windows present on the screen will be aligned.

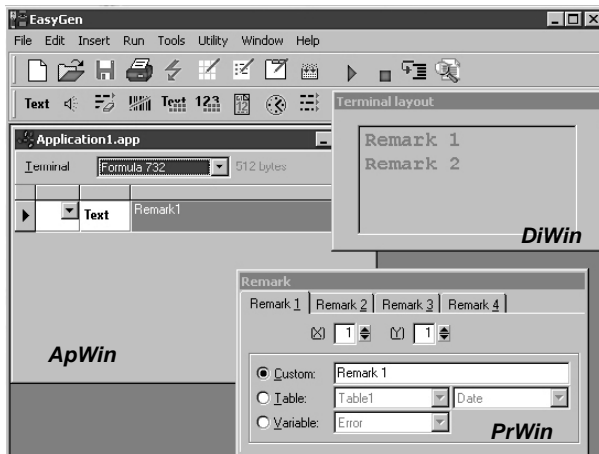
Formula®

5 INSERT MENU FUNCTIONS

5.1 REMARK FUNCTION

Text

Allows you to display on the Formula terminal up to four different comments in four locations.



PrWin has four tabs, Remark1, Remark2 ... Remark4, for four comments. DiWin shows the display status with the four comments.

The comment can be specified as a message from the programmer, a table field or the value of a variable.

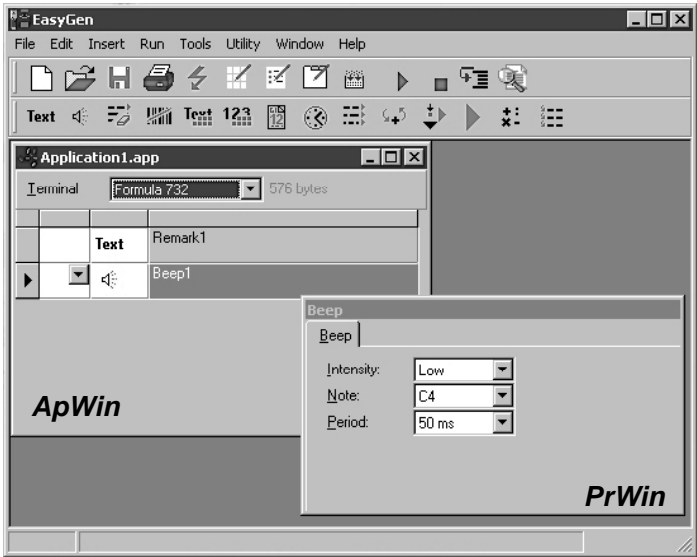
Acting upon (X) and (Y) it is possible to position the initial coordinates of the comment.

The result of the comment function is visible in the window that simulates the display of the Formula terminal.

5.2 BEEP FUNCTION



Allows the Formula terminal to emit a beep. Also it is possible to define its loudness, the pitch, and the duration on the PrWin.



5.3 CLEAR DISPLAY FUNCTION



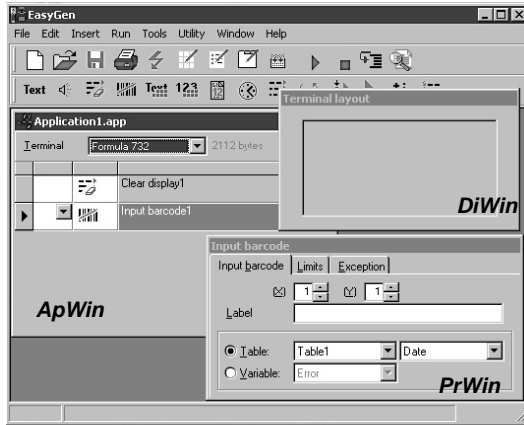
Completely clears the display of the Formula terminal.
PrWin and DiWin are not necessary.



5.4 INPUT BARCODE FUNCTION

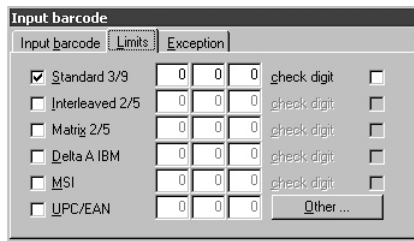


Allows you to read a bar code and save the information in the field of a table or in a variable. It is possible to display a predefined comment in any position on the screen. The code when read or keyed will be displayed after the last characters of the comment.



PrWin has three tabs:

- **Input barcode tab** for the description of the comment and the saving location of the code entered, a table field or a variable.
- **Limits tab** for the choice of bar code type and lengths to be read:



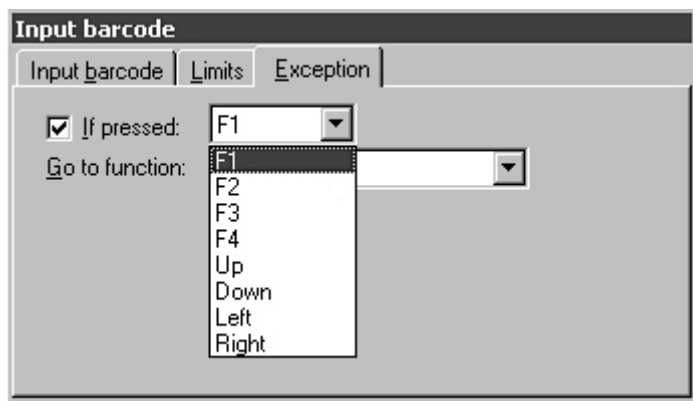
For each code type it is possible to define 3 acceptable lengths.

A value of 0 will accept any length. When there are only one or two acceptable lengths, it is necessary to fill the remaining boxes with the same value. In addition, it is possible to handle the control digit or not, as desired.

The <Other> button permits access to other types of codes:

- **Exception tab** where you can define an exit from the bar code acquisition mode by pressing a particular button, and continue with the program by jumping to a specified function.

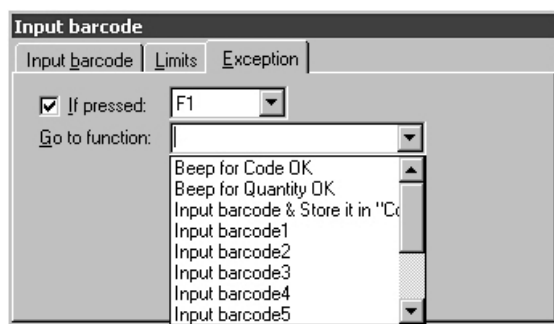
The available exit keys for the Formula 734 terminal are:



Other available Formula terminals may have more or less keys. See the relevant Formula terminal User's Manual.

**NOTE**

It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.



– **EAN128 & CODE 128**

By enabling CODE 128 codes, it is possible to read, not only normal CODE 128 barcodes, but also single field EAN 128 codes.

By enabling EAN 128 codes, it is possible to read, not only single field EAN 128 barcodes, but also CODE 128 codes.



NOTE

To correctly read EAN 128 codes of more than one field, it is necessary to create the special EAN128tb table. (See par. 4.6.1).

This table must have a number of fields greater than or equal to the maximum number of fields to be read.

The EAN 128 code content will ALWAYS be assigned to the EAN128tb table. The code fields will automatically be separated and saved in their respective fields defined in the table, always starting from the first field.

The EAN128tb table is made up of ONLY one record. Therefore it is necessary to save this data elsewhere before reading a new code (otherwise it will be overwritten).

Enabling multiple code types simultaneously

- If data is to be saved in the EAN128tb table, all other code types read (different from EAN 128) will be saved in the specified field of the EAN128tb table.
- If data is to be saved in a table other than EAN128tb (for example "table1"), all other code types read (different from EAN 128) will be saved in the specified table.

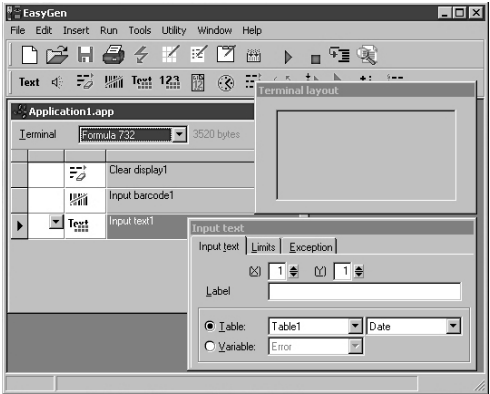
To test a program that allows reading EAN 128 codes using the simulator (see par. 4.5):

- use the "|" (Pipe) character as the first character of the code
- use the "|" (Pipe) character as the field separator (instead of character 'GS')
For example: "|ONE|TWO|THREE"

5.5 INPUT TEXT FUNCTION



This function allows you to enter a sequence of characters from the Formula terminal keyboard and to save the information in a table field or in a variable. It is possible to display a fixed comment in a pre-defined position.



To set the maximum number of acceptable characters, just click the **Limits** tab and enter the value. In your application, if more characters than the maximum acceptable value are input, the string of characters entered will be truncated to the maximum acceptable value.

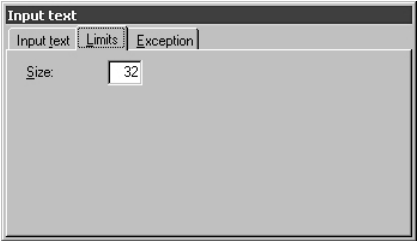
Example:

Size = 3

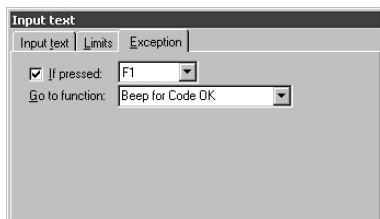
Input string = "ABCD"

When the letter "D" is input, the string "ABC" is automatically accepted.

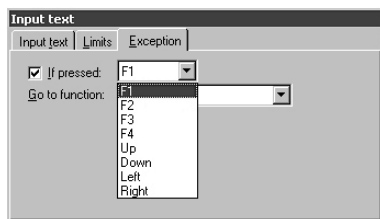
The maximum possible value is 32 characters (default).



With the **Exception tab**, you can define an exit from the text acquisition mode by pressing a particular button, and continue with the program by jumping to a specified function.



The available exit keys for the Formula 734 terminal are:

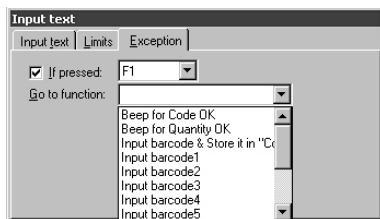


Other available Formula terminals may have more or less keys. See the relevant Formula terminal User's Manual.



NOTE

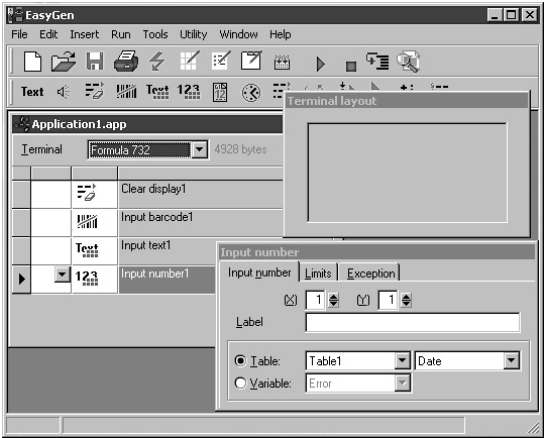
It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.



5.6 INPUT NUMBER FUNCTION



This function allows the user to insert a number and to save the information in a table field or in a variable. It is possible to display a fixed comment in a predefined position.



To set the maximum number of acceptable digits, just click the Limits tab and enter the value. In your application, if more digits than the maximum acceptable value are input, the string of digits entered will be truncated to the maximum acceptable value.

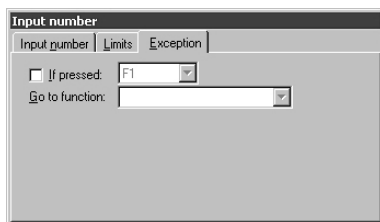
Example:

Size = 3
Input string = "1234"

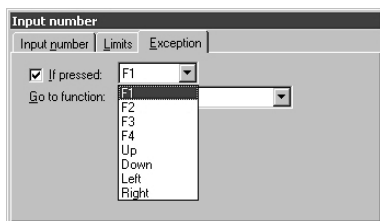
When the digit "4" is input, the string "123" is automatically accepted.
The maximum possible value is 32 digits (default).



With the **Exception tab**, you can define an exit from the number acquisition mode by pressing a particular button, and continue with the program by jumping to a specified function.



The available exit keys for the Formula 734 terminal are:

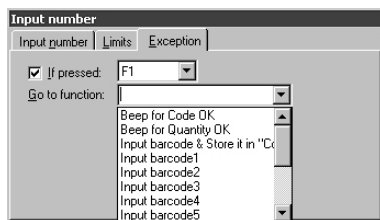


Other available Formula terminals may have more or less keys. See the relevant Formula terminal User's Manual.



NOTE

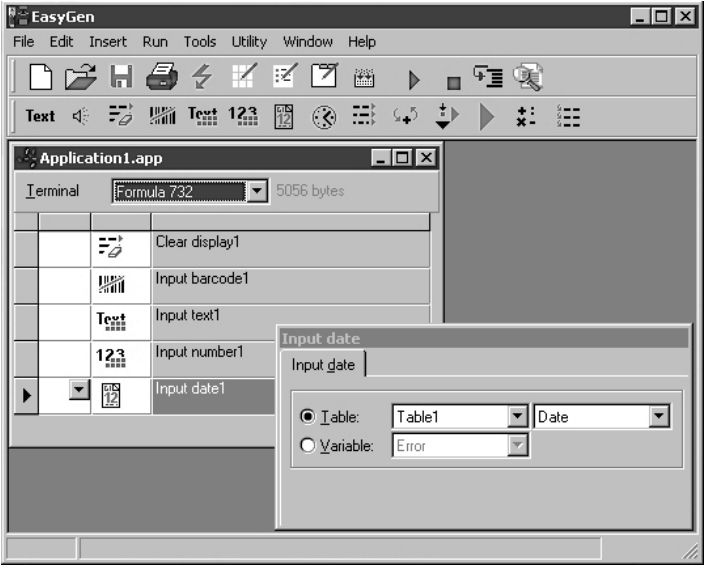
It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.



5.7 INPUT DATE FUNCTION



This function allows the user to save the system date in a table field or in a variable.

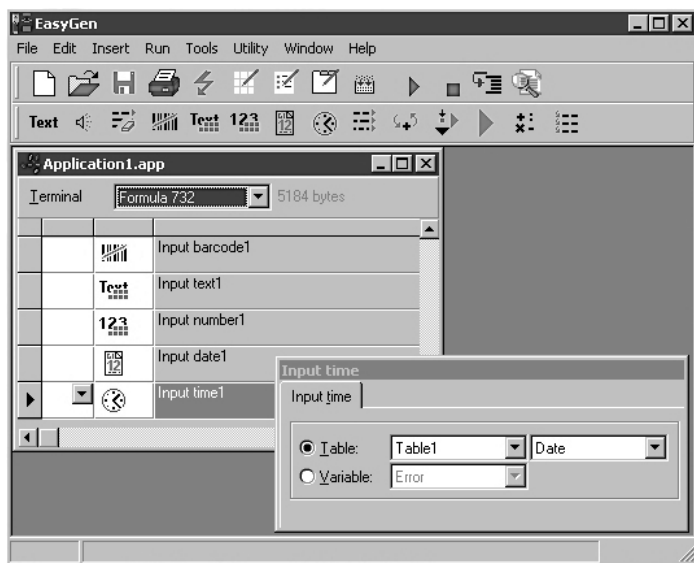


To avoid possible errors, it is recommended that the field in which you wish to save the date be defined as a “date” field (see par. 4.6.1).

5.8 INPUT TIME FUNCTION



This function allows the user to save the system time (handled by an internal clock in the terminal) in a table field or in a variable.

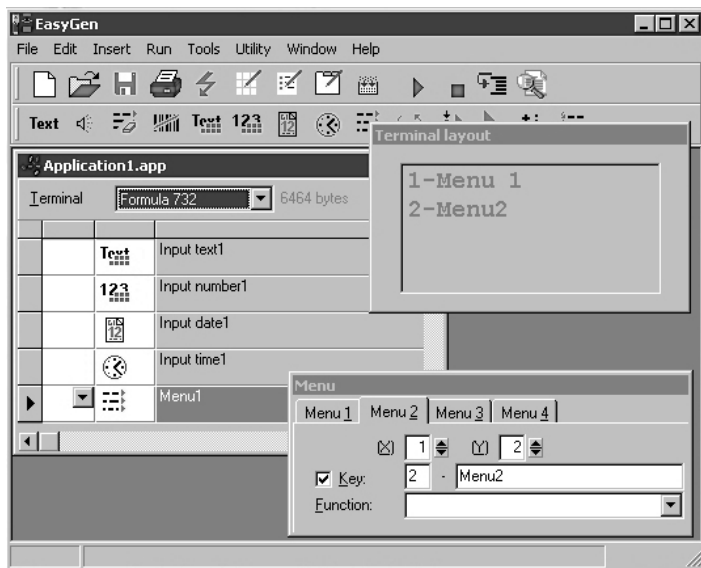


To avoid possible errors, it is recommended that the field in which you wish to save the time be defined as a “time” field (see par. 4.6.1).

5.9 MENU FUNCTION



Allows the creation of a menu with a maximum of 4 choices. For each choice it is possible to define the position, the key to press, the text, and the function to be executed.



The menu always uses a numerical key for the choice options.



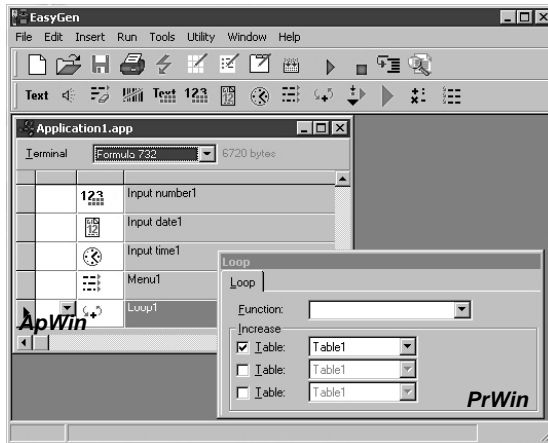
NOTE

It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.

5.10 LOOP FUNCTION



This function allows the user to increment the index of three tables and define a function loop which repeats until a specified condition is met.



The loop begins with the function defined with the parameters in the PrWin and ends with the function “Loop”. This loop is repeated until a jump is made to a function which allows an exit.

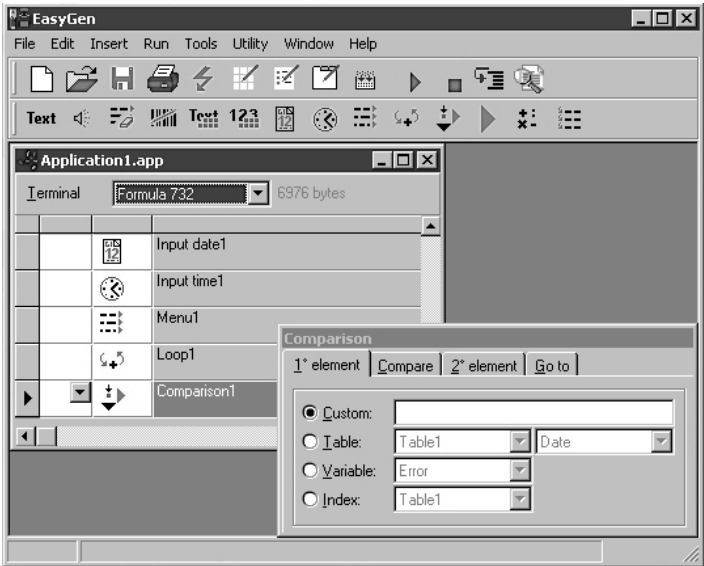
The first time the program is used, or when a table is re-loaded, the index is always pointing to the first record. The user can choose which index(es) to increment in the PrWin of the loop function.

If the index is not incremented, it always points to the same record, and therefore each successive code read will overwrite the previous one.

5.11 COMPARISON FUNCTION



The first element is compared with the second, and based on the result (true or false) a decision is made as to where the program continues.



The comparison can be made between numbers or character strings. Both the first and second elements may be constants or a table field.

In the case of numerical comparison, the constant must always be defined as the second element.



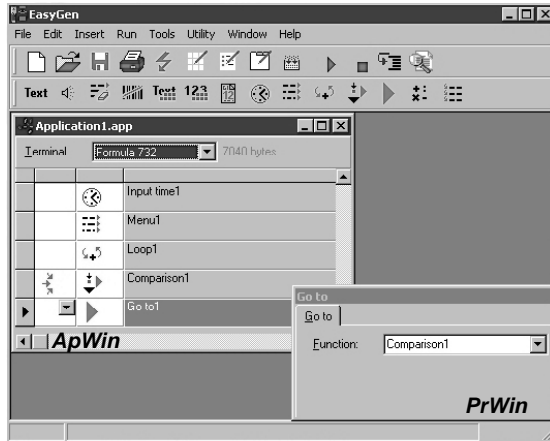
NOTE

It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.

5.12 GO TO FUNCTION



This function causes the program execution to jump to a pre-defined function. By using the down arrow in the PrWin , one obtains a list of all available functions.



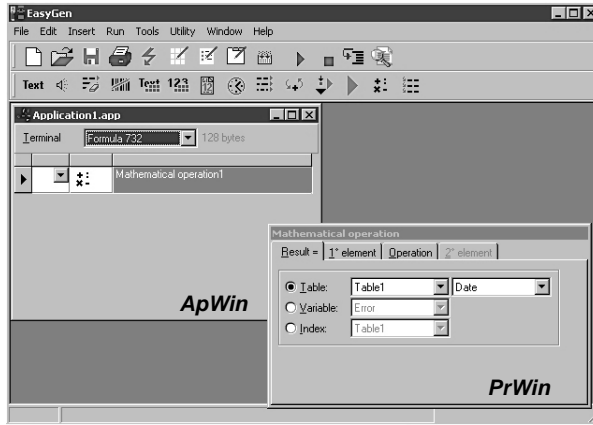
NOTE

It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.

5.13 MATHEMATICAL OPERATION FUNCTION

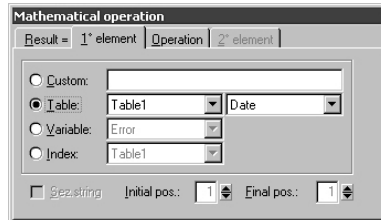


This function allows the execution of arithmetic operations or string operations, and assignments.

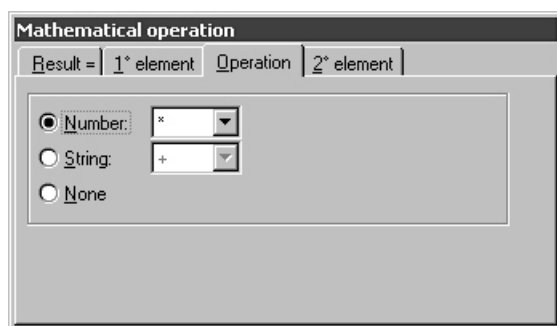


The “Mathematical operation” PrWin is composed of four tabs:

- **Result=** The result can be saved in:
a table field,
in a variable,
in the index of a table.
- **1° element** The first element of the operation may be:
a constant,
a table field,
the value of a variable,
the index of a table.

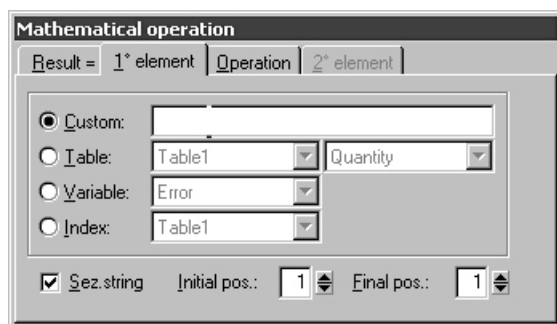


- **Operation** It's possible to define operations between number, string or none:
 - the *arithmetic* operations are the four base operations, addition, subtraction, multiplication, and division, and also percentage;
 - the available *string* operations are concatenation or substring. The concatenation is the union between the string contained in the first element with the string contained in the second element. Concatenation is selected in the Operation list box, item "+".



In this tab, if you check the Operation box "None", you can also define the substring operation.

The substring operation is selected in the "1° element" tab. You have to check the Operation box "None" and then, in the "1° element" tab you have to check the Sez. string box, and specify the first and the last position of the string required. The result string must be stored in a variable.



- The none operation copies the "1°element" into the location defined in the "Result=" tab.

- **2° element** The second element of an operation between numbers or strings may be:
 - a constant,
 - a table field,
 - the value of a variable,
 - the index of a table.In the case where the arithmetic operation is an assignment (operation = none), the second element must not be defined.

Mathematical operation

Result = 1° element Operation 2° element

☐ Custom:

☐ Table:

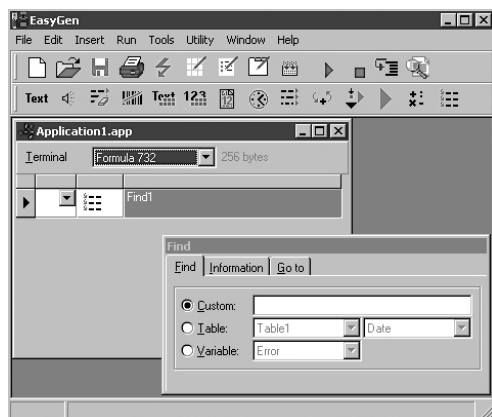
☒ Variable:

☐ Index:

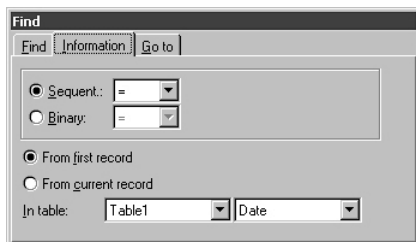
5.14 FIND FUNCTION



This function allows the user to search for a constant, the value of a table field or the value of a variable, within a table.



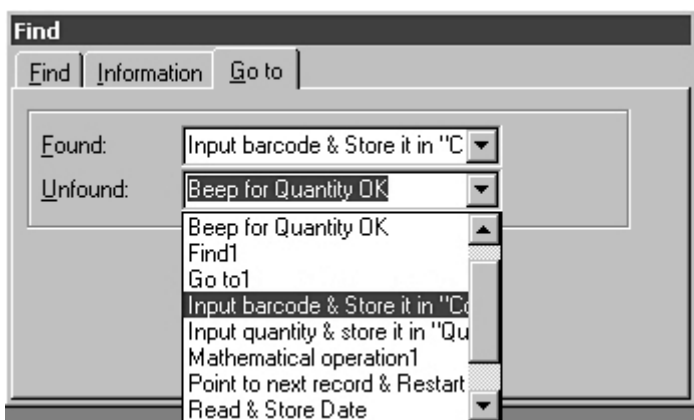
The Information option allows the user to define the type of search which can be sequential, or in the case of an ordered table, according to the field to be searched, with the binary method (also called “dichotomous”) and may begin with the first record or with the current record (index value).



To obtain the best results using the binary method, it is advised to use tables (relative to the search field) that are:

- completely full
- already ordered
- without duplicate elements

The Go to option defines the function to execute after the search.



Based on the results of the search, the program will jump to one of two functions specified.

The result of the search positions the table index at the record containing the field searched for if the search is of the equivalence type, or at the first record which satisfies the requested comparison.

If no record is found, the index variable maintains its current position.

**NOTE**

It is only possible to jump to functions which are already defined in the application. The creation of a jump to a function that has not yet been defined must be done by successively returning to this function menu.

Formula®

6 EASYGEN™ INTERPRETER PROGRAM

**NOTE**

In this chapter the images representing the Formula terminal are only used as examples to demonstrate use of the keys and messages appearing on the display.

6.1 INTERPRETER PROGRAM USE

Application programs created by EasyGen™ can be run on the Formula terminal only if the Elxxx_yyy.HEX interpreter program has been previously loaded onto the terminal.

Some example application programs are installed onto the PC and can be loaded onto the Formula terminal using either EasyGen™ (see par. 4.2.7) or DOSDL.EXE (see Appendix B).

6.2 USE OF THE FORMULA TERMINAL

Once the application software is loaded onto the Formula terminal (either an example supplied or an application developed with EasyGen™) to launch it, it is necessary to extract the Formula terminal from the cradle and press <SHIFT> and then the <UP ARROW> key.

If the barcode scanning function has been used in the application, scanning is possible.

6.2.1 Description of Keys

Although EasyGen™ allows assigning customized functions to function keys, there are some standard functions common to all applications because they are implemented in the interpreter software.



NOTE

Every time a key is pressed, the Formula terminal remains turned on for a maximum of 20 seconds.



ARROW KEYS: only available for the application when not in “Select or Data Edit” (see par. 6.2.3).



SCAN KEY: Enables the laser for barcode scanning and turns the terminal on if it is off.



FUNCTION KEYS: up to eight function keys are available with the following meaning:

Free for EasyGen™ Application	F734	<F1>, <F2>, <F3>, <F4>
	F732	<F1>, <F2>, <F3>, <F4>
	F725	<F1>, <F2>
	F660	<F1>, <F2>
Select Table	F734	<F5> = <SHIFT><F1>
	F732	<F5> = <SHIFT><F1>
	F725	<F5> = <SHIFT><SHIFT><UP ARROW>
	F660	<F3> = <SHIFT> <F1>
Data Search	F734	<F6> = <SHIFT> <F2>
	F732	<F6> = <SHIFT> <F2>
	F725	<F6> = <SHIFT><SHIFT><DOWN ARROW>
	F660	<F4> = <SHIFT> <F2>
Deletion	F734	<F7> = <SHIFT> <F3>
	F732	<F7> = <SHIFT> <F3>
	F725	<F7> = <SHIFT><SHIFT><LEFT ARROW>
	F660	<F5> = <SHIFT><SHIFT><F1>
Data Display	F734	<F8> = <SHIFT> <F4>
	F732	<F8> = <SHIFT> <F4>
	F725	<F8> = <SHIFT><SHIFT><RIGHT ARROW>
	F660	<F6> = <SHIFT><SHIFT><F2>



ESC KEY: used in the “Data display” mode (see 6.2.3).



SHIFT KEY: Enables the entry of alphabetical characters (written in white on the keyboard) when followed by the pressing of a numeric key: for example if you want to enter the alphabetical character “A”, you have to press <SHIFT>+<7>. The number of times the SHIFT key is pressed determines the choice of alphabetical character: for example if you want to enter the alphabetical character “N”, you have to press <SHIFT>+<SHIFT>+<5>, if you want to enter the alphabetical character “X”, you have to press <SHIFT>+<SHIFT>+<SHIFT>+<2> and so on. The fourth time the SHIFT key is pressed, the SHIFT function is disabled. The SHIFT function can also be disabled by waiting for a time-out of 2 seconds after being pressed.



By pressing the <SHIFT> key followed by the <RIGHT ARROW> key, the graphic display's contrast increases to the allowed maximum, and then returns to 0 value.



By pressing the <SHIFT> key followed by the <LEFT ARROW> key, the back-lighting will be turned on/off. This function is **not** available in **F732** terminal.



ALPHANUMERIC KEYBOARD: Enables the numeric characters to be entered and displayed. If the <SHIFT> key was pressed previously, the choice of alphabetical character (written in white on the keyboard) is enabled.



BACKSPACE KEY: deletes the last character entered.



SPACE KEY: Introduces a space. If the <SHIFT> key was pressed beforehand, the choice of optional characters is made possible.



ENTER KEY: allows validation of what has been typed.



SHIELDED RESET KEY: it is activated by inserting a blunt object in the slot while simultaneously pressing the <SCAN> key (see par. 6.3).

6.2.2 Select Table

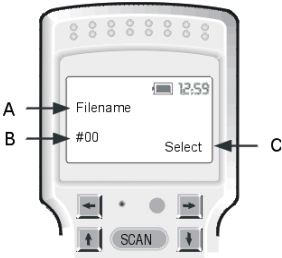
To activate the "Select" table function press the keys according to the Formula terminal in use as suggested in the following table:

Select Table	F734	<F5> = <SHIFT><F1>
	F732	<F5> = <SHIFT><F1>
	F725	<F5> = <SHIFT><SHIFT><UP ARROW>
	F660	<F3> = <SHIFT> <F1>

All available files appear on the Formula terminal's display screen.

Key:

- A) File name
- B) File number (00..15)
- C) Active mode



By pressing the <UP ARROW> and the <DOWN ARROW> keys, it is possible to select one of the 16 possible files. File 15 is the application software file developed by the user with EasyGen™.

Once the correct file is selected, it is possible to choose the type of operation to execute by using the function keys (see par. 6.2.1 for details).



ATTENTION

File 15 (the application software file) must never be deleted.

To leave "Select" table, simply press the <ESC> key.
The last file selected continues to be displayed.

6.2.3 Data Display

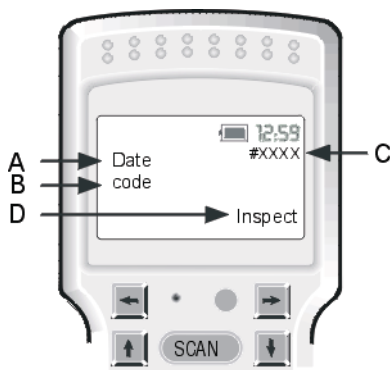
Once a table is selected, it is possible to view its contents on the Formula terminal display screen by pressing the keys according to the Formula terminal in use as suggested in the following table:

Data Display	F734	<F8> = <SHIFT> <F4>
	F732	<F8> = <SHIFT> <F4>
	F725	<F8> = <SHIFT><SHIFT><RIGHT ARROW>
	F660	<F6> = <SHIFT><SHIFT><F2>

The figure indicates what appears on the display screen.

Key:

- A) Field name
- B) Field contents
- C) Record position
- D) Active mode



It is possible to move in all directions by using the arrow keys, as indicated in the figure below.

	Code	Date	Time	Qty
Record n _i 0001	015910 →	11/09/1996 →	10:15 ↓	22
Record n _i 0002	015911	12/09/1996	11:22 →	15
Record n _i 0003	015912	12/09/1996	11:25	69 ↑

During the data display phase, it is possible to delete a field, modify it or proceed to a search function at any time.

6.2.4 Data Deletion

Two types of deletions are possible while using the Formula terminal:

- the deletion of the contents of a field
- the deletion of a data table.

To activate this function press the keys according to the Formula terminal in use as suggested in the following table:

Deletion	F734	<F7> = <SHIFT> <F3>
	F732	<F7> = <SHIFT> <F3>
	F725	<F7> = <SHIFT><SHIFT><LEFT ARROW>
	F660	<F5> = <SHIFT><SHIFT><F1>

To delete a field, it is necessary to display it on screen and to press the function keys as described above. A message requesting confirmation appears on the display screen.

The deletion request is confirmed by pressing the <ENTER> key; the contents of the field are deleted definitively.

Pressing the <ESC> key, on the other hand, cancels the deletion request; the field's contents will not be deleted.

Deleting the contents of a table selected on the display screen can be carried out by pressing function keys as described above. A message requesting confirmation appears on the display.

The deletion request is confirmed by pressing the <ENTER> key; the table is deleted.

Pressing the <ESC> key, on the other hand, cancels the deletion request; the table will not be deleted.



**NOTE**

The deleted file will not be eliminated from the Formula terminal; all field contents are deleted and file pointer is set to the first record.

6.2.5 Data Search

To activate this function press the keys according to the Formula terminal in use as suggested in the following table:

Data Search	F734	<F6> = <SHIFT> <F2>
	F732	<F6> = <SHIFT> <F2>
	F725	<F6> = <SHIFT><SHIFT><DOWN ARROW>
	F660	<F4> = <SHIFT> <F2>

It is possible to search for a code only if a data table has been previously selected.

The figure indicates what appears on the display.

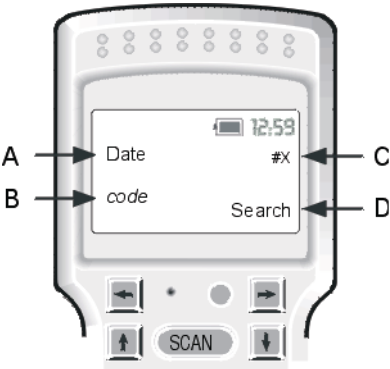


It is necessary to insert the code to be searched for and to press <ENTER>; what is shown in the figure will appear on the display screen.



If the code is found, what is shown in the figure will appear on the Formula terminal's display screen.
Key:

- A) Field name
- B) Field contents
- C) Record position
- D) Active mode



Once a code is found, it is possible to continue searching toward the beginning of the data table <UP ARROW> or toward the end of the data table <DOWN ARROW>. If other fields with that code are present, they will be displayed.

Once the code searched for is found, it is possible to modify its contents simply by inserting a new value and confirming it pressing <ENTER>. By doing this, "Edit" mode is entered and the figure indicates what appears on the display screen.



If the number of characters entered is less than the length of the field, it is filled by a series of zeroes (0).

If the code is not found, the figure indicates what appears on the Formula terminal's display screen.



Press <ESC> to exit the "Search" mode.

**ATTENTION**

There should not be more than three consecutive empty records during the data search phase, otherwise the current operation will be terminated.

6.2.6 Error Messages

If the Elxxx_yyy.HEX program detects an error during the execution of the application, the figure indicates what appears on the Formula terminal display.



Elxxx_yyy.HEX can detect eight possible types of errors:

ERROR TYPE	DESCRIPTION
DBASE FULL	Database full. During a SAVE or CHECK operation the data could not be written to the database or the record pointer reached its maximum limit.
INSTR. ERR	Wrong program instruction.
TIMER FLT	Internal error. EasyGen™ malfunction.
READ ERROR	Internal error. EasyGen™ malfunction.
UNK. ERROR	Unknown or generic type of error.
CMD NOT LD	Application program not present.
ACT. ERR	Wrong action.
FNF	File not found.

The “L” variable gives the line number containing the error, while the “S” variable indicates the type of operation that caused the error.

The possible values of the “S” variable and the corresponding error causes are indicated in the following table.

S	FUNCTION WITH ERROR	REMEDY
0	Acquisition	<p>The causes of this error could be:</p> <ol style="list-style-type: none"> 1 Non-permissible acquisition executed 2 Field acquisition of a non existent table executed. 3 Field acquisition executed where addressed record does not exist in table. 4 Next instruction does not exist.
1	Check	<p>Error causes are:</p> <ol style="list-style-type: none"> 1 Table to check does not exist.
2	Decode	Command containing an unexpected decode sent to terminal.
3	Enable Decode	Next instruction does not exist.
4	Sequential search	<p>Possible causes are:</p> <ol style="list-style-type: none"> 1 Table to check does not exist. 2 Comparison operator indicated not permissible.
6	Data Element	<p>Possible causes are:</p> <ol style="list-style-type: none"> 1 Error in current instruction. 2 Next instruction does not exist. 3 General user program error.
7	Pointer Element	<p>Possible errors are:</p> <ol style="list-style-type: none"> 1 Points to a non existent table. 2 Next instruction not among expected ones.
8	Search Type	<p>Only two types of searches are possible:</p> <ol style="list-style-type: none"> 1 BINARY: where the table must be ordered in ascending order in the field in question. 2 SEQUENTIAL: where the table may have any order.
9	Comparison Type	<p>Only two types of comparison are possible:</p> <ol style="list-style-type: none"> 1 With database field. 2 With constant.

S	FUNCTION WITH ERROR	REMEDY
11	Save	Two types of errors are possible: 1 Table does not exist. 2 Tried to save a data item in a non existent location.
29	Divisible Data Item	Next instruction does not exist.
30	Operation on Index	Four types of errors are possible: 1 Non existent table. 2 Wrong operator: only incremental and decremental are possible.
35	Go to Row	Addressed row does not exist.
36	Define Beep	Next instruction does not exist.
38	Binary Search	Table does not exist.
42	Display	Three errors are possible: 1 Table does not exist. 2 Operating system error. 3 Next instruction does not exist.
43	Start	Program does not exist.

6.3 DELETION OF THE APPLICATION PROGRAM



CAUTION

The procedure described below will entail the deletion of the data acquired and of the application software even if the procedure is not completed.

If the application is to be changed, it is necessary to first delete the application currently in use as indicated below.

- 1- Press <SCAN> and the protected RESET key simultaneously; the display screen will show:



- 2- To proceed with the deletion, press the ▲ key; after a few seconds, the display screen will show:

Now, you can load another application program using either EasyGen™ (see par. 4.2.7) or DOSDL.EXE (see Appendix B).



The deletion of the application program can be also obtained by reading the following ERASE ALL code:



ERASE ALL

- \$ % +



NOTE

By responding with a ▼, the interpreter software will also be deleted as indicated in par. 6.4.

6.4 DELETION OF THE INTERPRETER PROGRAM

If it is necessary to delete the interpreter software currently in use, proceed as follows:

- 1- Press <SCAN> and the protected RESET key simultaneously; the display screen will show:



If you wish to delete the interpreter program, press the ▼ key.

The terminal's LED will remain red until the end of the deletion operation. The terminal's display screen will show below.



When the terminal is inserted in the cradle.



Pressing the <SCAN> button when the terminal is not inserted in the cradle.

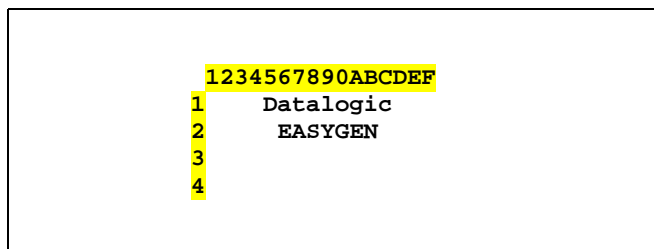
You can reload the interpreter program using either EasyGen™ (see par. 4.7) or DOSDL.EXE (see Appendix B).

Formula®

A PROGRAMMING EXAMPLES

Once the Formula terminal is inserted in the relative cradle, it is possible to see the courtesy message of EasyGen™ on the display.

The message can be set from the Option item in the Tools menu and it must have a maximum length of 4 lines, where each line can contain up to 16 characters (see below):



Then, you can load the following examples describing standard applications for portable Formula terminals:

- data collection/inventory
- sales aids
- loading/unloading lists.

These five examples take advantage of all the EasyGen™ functions, demonstrating the use of basic functions and the possibility of applying them directly in real projects.

For each example, the "new" functions will be explained in detail, leaving out the explanations for the more familiar and more often-used ones.

It is advised that these pages should be at hand when studying the applications directly.

A.1 EXAMPLE 1: CODE+QUANTITY

The first simple example, for inventory applications, allows the acquisition of barcodes, the setting of a quantity per code and the saving of everything inside the Formula terminal in a table. An internal calendar clock in the terminal also saves the date and time of the acquisition.

This program is totally linear, i.e. it is run from beginning to end without jumps or interruptions, as will be seen from the more complex examples below.

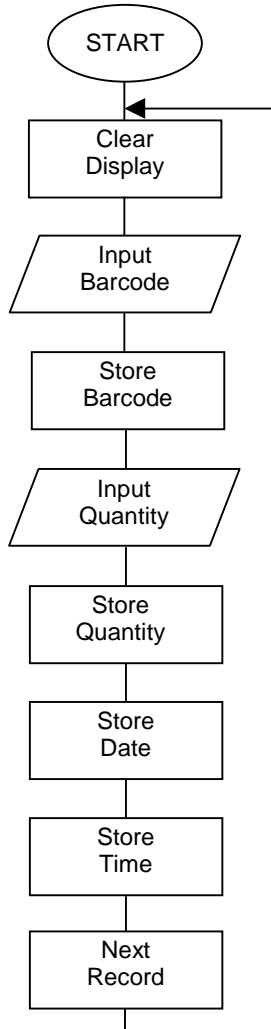
To understand how the examples functions, the first thing to check is the composition of the data table. In the EDITOR TABLES item of the TOOLS menu there is a setting selected for TABLE1 which is used to save all the scans.

There are 4 fields: DATE - TIME - CODE - QUANTITY

The Date and Time fields are automatically acquired from the terminal's internal clock via dedicated EasyGen™ functions.

The Code and Quantity fields, on the other hand, will be requested from the user.

❑ **Logic flow description**



Here is a list of all messages which can be displayed on the Formula terminals:

Start & Clear Display deletes the entire display so that each time the program starts anew, there is no leftover data from previous use that might disturb normal viewing.

This function has no parameters to be configured.

Input Barcode & Store it in “Code” Field stands by for the scanning of a barcode whose specifications are indicated between configurable parameters. After the scan, the data is saved in TABLE1 in the Code field.

Beep for OK code emits an acoustic signal to confirm the code scanning.

Input Quantity & Store it in “Quantity” Field stands by for the quantity reading (numeric value).

After the scan, the data is saved in TABLE1 in the Quantity field.

Note that among the settings under LIMITS the SIZE value is set to 5 or the maximum number of acceptable characters defined in the Quantity field description in TABLE1. If this value is left at 0 (default) during input, more than 5 characters will be accepted but the data will not be recorded and the Error variable will assume the value of 31 to indicate this.

Beep for Quantity OK emits an acoustic signal to confirm the quantity inserted.

Read & Store Date saves the date in the Date field, acquiring it in real time from the Hardware clock.

Read & Store Time saves the date in the Time field, acquiring it in real time from the Hardware clock.

Point to next record & Restart allows the program to restart and so request a barcode and a quantity once more.

❑ **Sample barcodes**



Type: code 39



Type: Interleaved 2/5



Type: Code 128



Type: Ean 13

A.2 EXAMPLE 2: EAN128 3 FIELDS

This is a special version of EXAMPLE 1 so please refer to that example for general specifications.

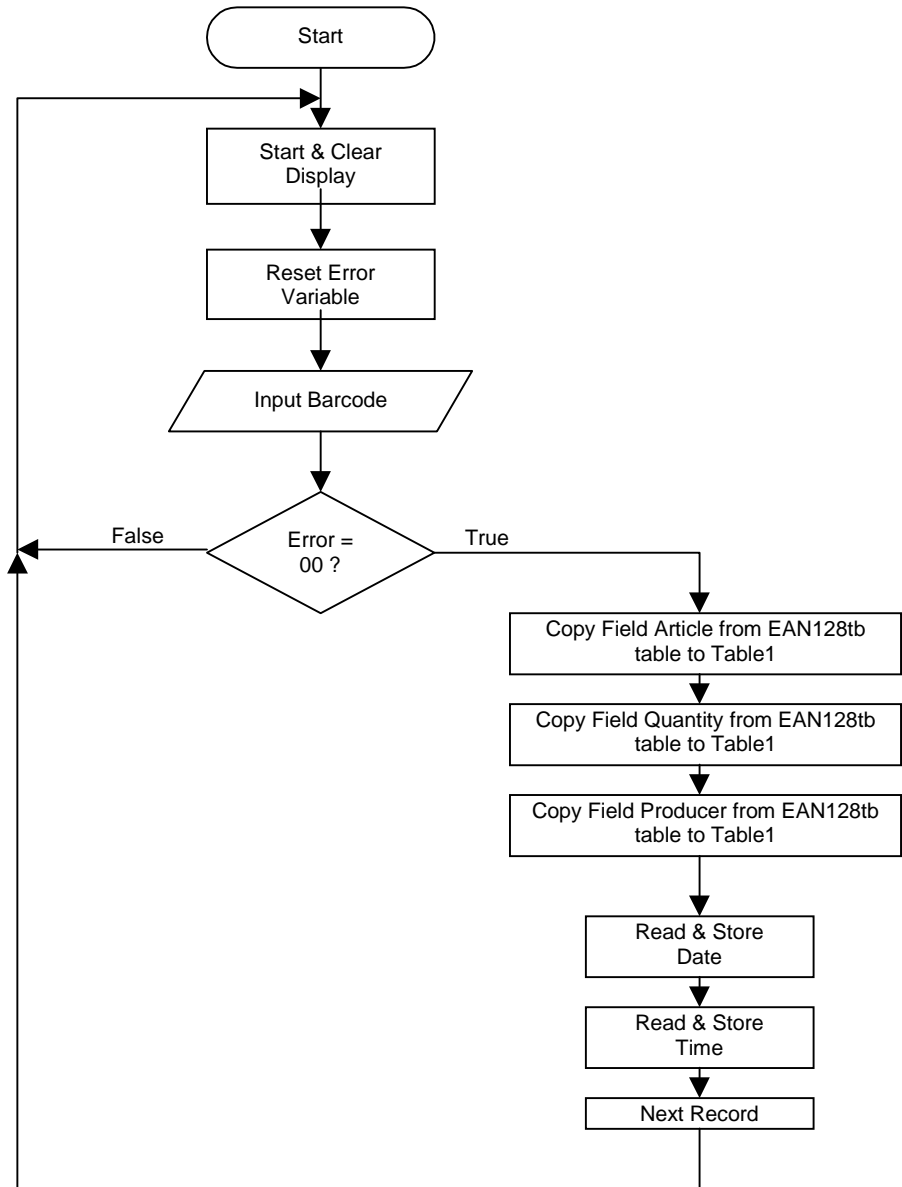
The example demonstrates how the use of code EAN128 can simplify repetitive operations which often take place in the warehouses.

This code gives information about the article, its producer and its quantity, which the user can get through a single read of the barcode label. This is very useful when several pieces of the same article are shipped together in the same box, as it allows to get all information required by reading the barcode of just one piece.

Through EasyGen™ it is possible to select the special table EAN128tb (see par. 4.6.1), which allows to read the code fields. Then, it is necessary to download the acquired data from the EAN128tb to TABLE 1.

There are five fields: ARTICLE - QUANTITY - PRODUCER - DATE - TIME.

❑ **Logic flow description**



Reset error sets the error variable to zero to check that the scanned code is correct.

Error = 00? compares the error variable to the value 00 to check if it is correct.

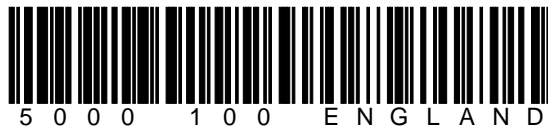
Input barcode & store it in EAN128tb saves the data in the EAN128tb table.

Store code in "Table1" field "Article" saves the data in TABLE1 in the Article Field, acquiring it from the EAN128tb.

Store code in "Table1" field "Quantity" saves the data in TABLE1 in the Quantity Field, acquiring it from the EAN128tb.

Store code in "Table1" field "Producer" saves the data in TABLE1 in the Producer Field, acquiring it from the EAN128tb.

❑ **Sample barcodes**



A.3 EXAMPLE 3: ASSISTED SALES

This example shows how all EasyGen™ options work.

The idea is to assist sales by referring to an item database inside the Formula terminal. Thanks to a customized menu, it is possible to organize the client management and get an on-line total cost for him.

Two ASCII files must be created (with any DOS-Windows ASCII compatible text editor). The first file is INPUT.DAT, which contains the article codes of the warehouse and gives them an item description and unit price. The second one is CLIENT.DAT, which contains a reference table with a Client Code and a Client Description.

The program starts by offering a personalized menu through the MENU function, which allows the creation of a series of customizable items associated with a keyboard character (see par. 5.5).

Two choices are available: the client setting and the cost setting. If the client setting is selected, the program will refer to the CLIENT.DAT table. In this way all the following "cost" readings will refer to that particular client and a client code field appears in the output table.

The code scanning allows to verify the presence or lack of the code in the warehouse and get the item description, its unit price, and (after entering the item "quantity found") its total cost.

The "search functions" are used in the example to search for the code in the internal database, while the "mathematical functions" allow to calculate the total costs.

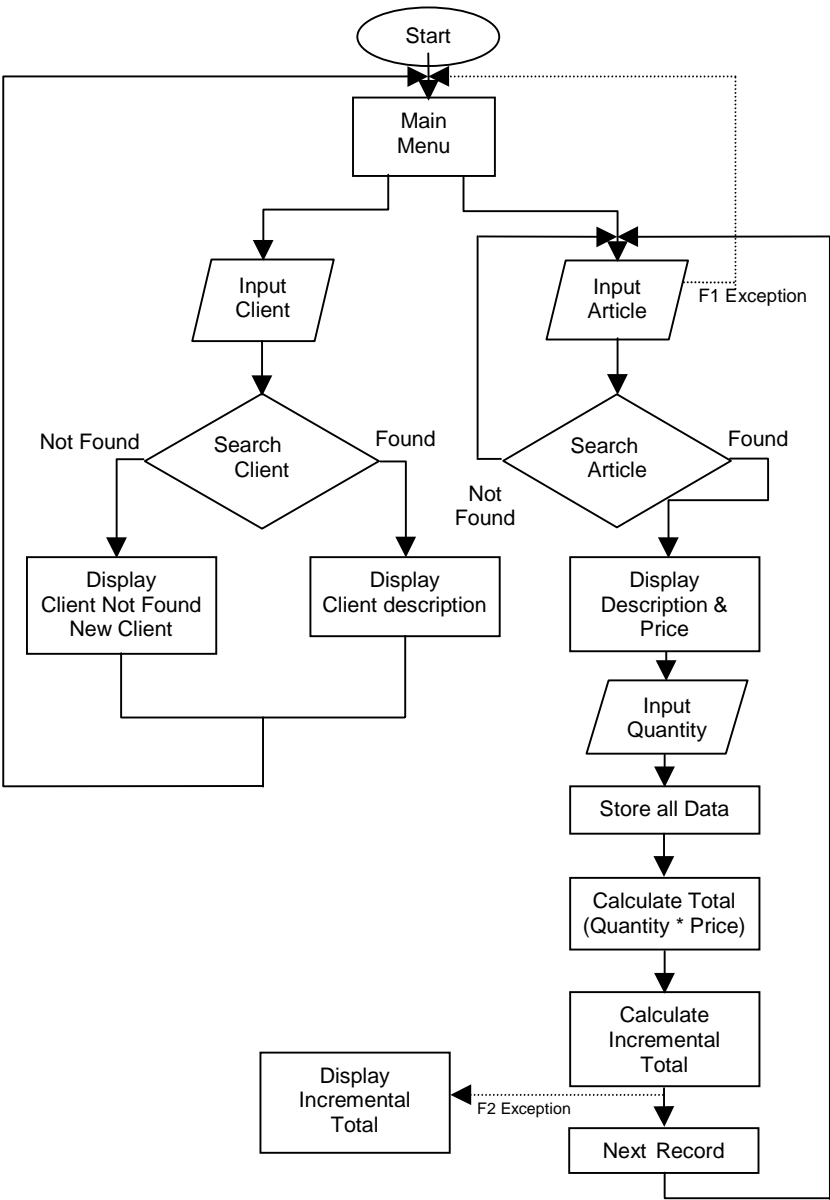
There are two exceptions. The first one occurs when the article is requested. In this case the program goes back to the main menu by pressing F1. This allows to close the cost for a client and to go on with a new client's setting.

The second exception regards the progressive calculation of the total. The Increase variable sums all the separate costs. Then, by pressing F2 instead of Enter to continue, it is possible to see on the display a client total cost up to that moment. This does not mean that the sale is closed, rather that it can be displayed (check) at all times.

There are three Tables:

- INPUT TABLE: contains the field structure of the code reference table (created with DOS edit command) with the item description and its price. The file should be called INPUT.DAT and its fields are: ARTICLE - DESCRIPTION - PRICE.
- CLIENT TABLE: contains the field structure of the client reference table (created with DOS edit command) with the client code and the client description. Its fields are: CLIENT CODE - CLIENT DESCRIPTION.
- OUTPUT TABLE: contains the field structure of the output table (data downloading): DATE, TIME, CODE, QUANTITY, TOTAL.

Logic flow description



Menu prompts the "define client" option OR the "shopping" (article) one.

Input barcode article stands by for the scanning of a barcode whose specifications are indicated between configurable parameters.

After the scan, the data is recorded in the Article variable. The choice of the variable originates from the fact that before saving it in the INPUT table, it is necessary to check that the code scanned is present in the codes reference table.

Search for article in input database is used to search for the code that has just been scanned and recorded in the Article variable in the INPUT table. Depending on the outcome (found or not found), the program will proceed to different phases.

Press Enter 0 allows to go on with the program.

Found - Display description & price is called when the article is found in the database of the INPUT table. It allows to display the item description and the unit price.

Input quantity stands for the article quantity, which is directly recorded in the Quantity field of the OUTPUT table.

Calculate & Store total this function is a real mathematical operation. The total is calculated by multiplying the Price field of the INPUT table by the Quantity field of the OUTPUT table. The product is recorded in the Total field of the OUTPUT table.

Calculate incremental total this function is a real mathematical operation. The current total is added to the previous incremental total. The amount is recorded in the Incremental variable.

Input barcode client stands by for the scanning of a barcode whose specifications are indicated between configurable parameters.

After the scan, the data is recorded in the CLIENT table.

Search for client in the CLIENT table is used to search for the client that has just been scanned and recorded in the CLIENT table. Depending on the outcome (found or not found), the program will proceed to different phases.

Client found - Display description is called when the client is found in the CLIENT table. It allows to display the client description.

❑ **Sample barcodes**

ARTICLE

F734



CLIENT

DATALOGIC QP NETWORK ITALY



F732



DATALOGIC QP NETWORK FRANCE



F725



DATALOGIC QP NETWORK SPAIN



ARTICLECLIENT

F660



DATALOGIC QP NETWORK GERMANY



F630



DATALOGIC QP NETWORK U.K.



MW25



A.4 EXAMPLE 4: LISTED PICKING

This example demonstrates that the INPUT files can also be used as OUTPUT files after processing by the terminal.

A LIST table is loaded that refers to the LIST.DAT ASCII file containing a series of articles to be taken.

The operator is told which code to take, its position and the quantity to be taken.

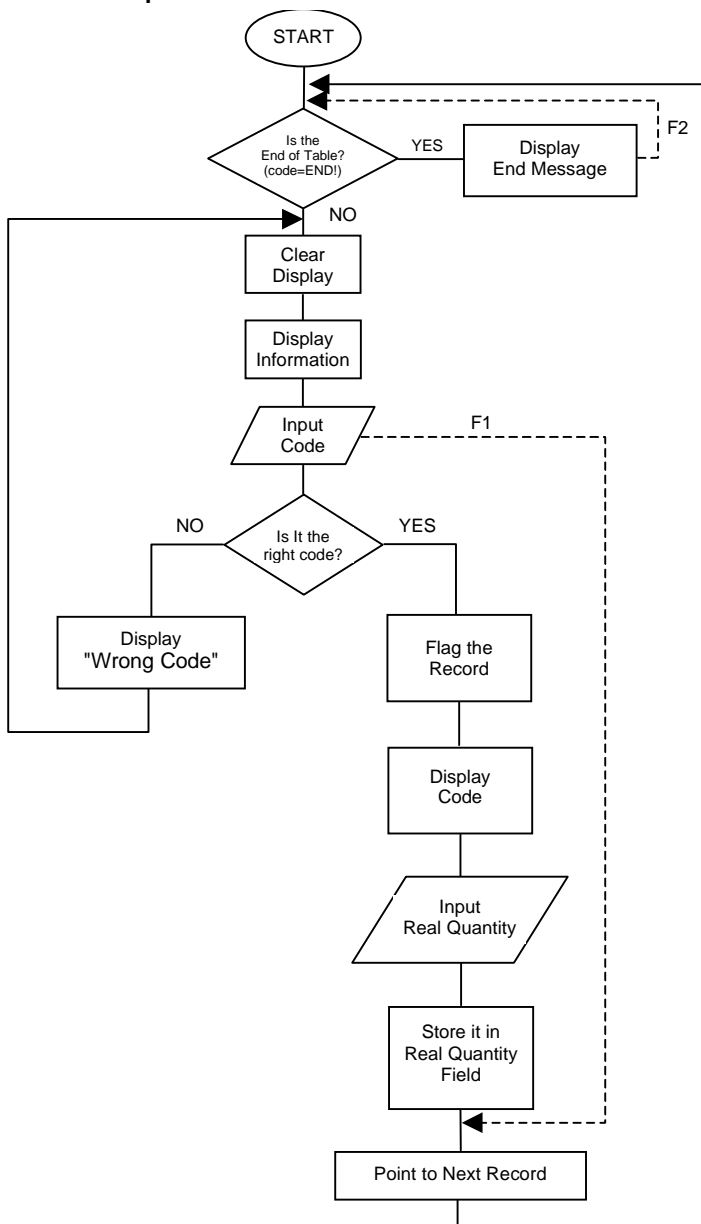
After checking that the code read ,matches the one requested, the exact quantity to be taken is requested and the processed record is marked with an "X" in the FLAG field.

If you press F1 before selecting a code, it is possible to jump to the following code of the LIST table. In this way the ignored code will not be flagged.

At the end of the loading list, checked by setting an "END! Code", the operator is asked to download the processed table by using SysTools™2001 program (see SysTools™2001 User's Manual).

Once the table is downloaded, it is necessary to delete it before creating a new one. Then, the new table is sent to the interpreter. By clicking the F2 button on the terminal it is possible to start again.

❑ Logic flow description



❑ **Sample barcodes**



A.5 EXAMPLE 5: INVENTORY

This example allows the creation of an inventory of the articles and their relative quantity.

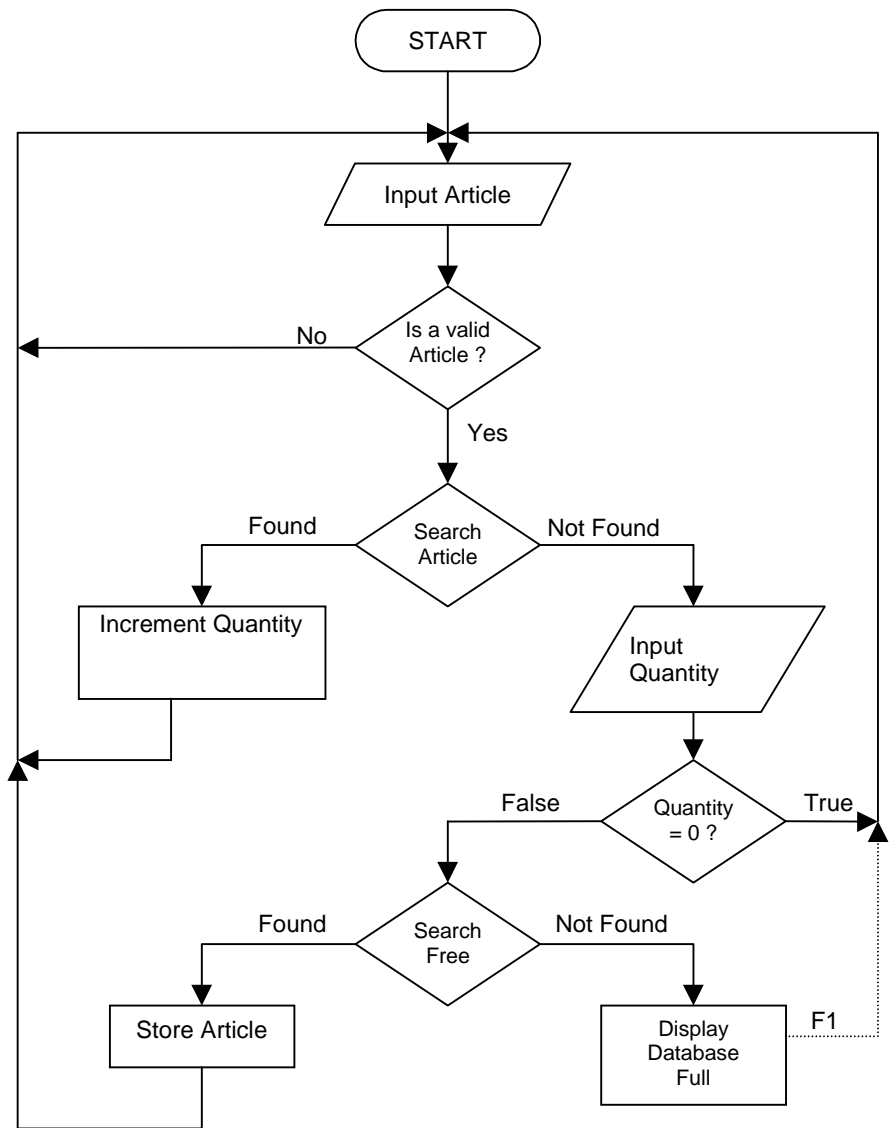
When entering an article code, the program EasyGen™ will automatically search for this article in the database. If the article is found, its quantity will be increased by 1; if it is not found, it is recognized as a new article and recorded in a new position. In this case it is required its initial quantity.

When an article quantity corresponds to 0, the program does not record it. In this way it possible to avoid the record of a wrong article in the database.

While recording new data, it is possible to see the current inventory on the display through the function keys (see par. 6.2.1). At the end of the operation the inventory can be downloaded in a table with SysTools™2001.

The database is recorded in the ARTICLE table, which can contain up to 500 records and it has two fields: CODE - QUANTITY.

❑ Logic flow description



☐ **Sample barcodes**

Formula[®]

B DOSDL

B.1. LOADING FILES USING DOSDL.EXE

DOSDL.EXE is a utility program that allows the loading of files to the Formula terminal under the DOS environment (DOS 5.0 or higher).

The following files can be loaded:

- EasyGen[™] Interpreter program
- application files with associated tables exported by EasyGen[™]
- individual tables.

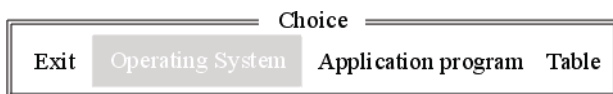
DOSDL is placed in the installation directory of EasyGen[™] on your PC.

After making the RS-232 connection between the Formula terminal and the host computer, execute DOSDL.EXE by entering the following at the DOS prompt:

DOSDL	if the COM1 serial port is used;
DOSDL /L2	if the COM2 serial port is used.

It is possible run the command "DOSDL /?" to display the help.

The DOSDL “Choice” menu appears on the host computer monitor:

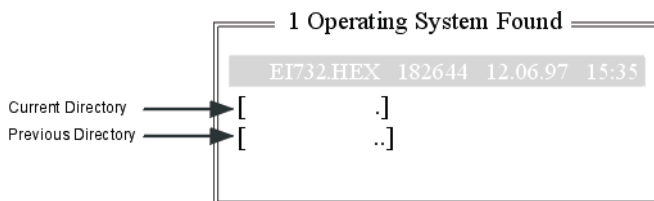


From this menu it is possible to select the item relevant to the file that is to be loaded onto the Formula terminal.

B1.1. DOSDL.EXE - Interpreter Program

To load the Interpreter Program in the Formula terminal, the item “Utilities - Initiate...” in EasyGen™ main menu can be used, or the “Operating System” item can be selected from the DOSDL “Choice” menu, and the <ENTER> key pressed.

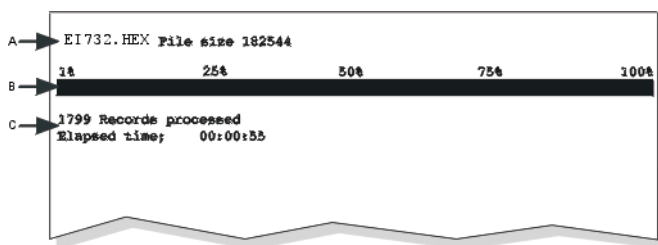
A list of all the interpreter programs (“HEX” extension) present in the current directory appears on the monitor (it is possible to change directories):



ATTENTION

It is necessary to change the name of the interpreter program file (EIXXX_YYY.HEX) since it must have a maximum length of 8 characters while working in a DOS environment..

Select EIXXX_YYY.HEX and press <ENTER>, and the following appears on the monitor:



Key:

- A) File name and size
- B) Loading status bar
- C) Reply message

To interrupt the installation of the EasyGen™ Interpreter Program at any time, just press <CTRL>+<BREAK> from the computer keyboard.

It is also possible to load the EasyGen™ Interpreter Program using DOSDL.EXE directly from the DOS prompt. Enter the command:

DOSDL / EIXXX_YYY.HEX

if the COM1 serial port is used;

DOSDL / L2 / EIXXX_YYY.HEX

if the COM2 serial port is used.

When the Interpreter Program has finished loading, the Formula terminal begins a memory test. The display screen shows what is indicated in the figure.

In case of error detected during the running of the test, the Formula terminal freezes. In this case, it is advisable not to run other operations and to contact a technician.



If no errors are detected during the running of the test, the Formula terminal is able to interpret the application commands created with EasyGen™ and the display screen shows what is indicated in the figure.



The interpreter program EIXXX_YYY.HEX can be deleted just like any other application.



ATTENTION

The activation of the reset button always causes the deletion of data and application programs, which must then be reloaded.

The reading of ERASE ALL causes the initialization of the interpreter program with deletion of the data and application program (see par. 6.3)

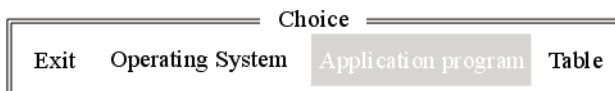
The EasyGen™ Interpreter for Formula 732 does not handle the RETURN TO DEFAULT PARAMETERS barcode.

B1.2. DOSDL.EXE - Application program

To load the application in the Formula terminal, the item “File – Send” can be selected from EasyGen™ main menu, or the DOSDL.EXE program can be used.

First of all it is necessary to use the item “File - Export” in the EasyGen™ main menu to prepare the files required by DOSDL for application loading. Place any table.DAT files required by your application in the same directory as the “Export” saved files.

After making the RS-232 connection between the Formula terminal and the host computer and loading the application onto the Formula terminal, it is necessary to select the “Application program” item from the DOSDL “Choice” menu:



A list of all applications ("EXP" extension) present in the current directory appears on the monitor (it is possible to change directories):

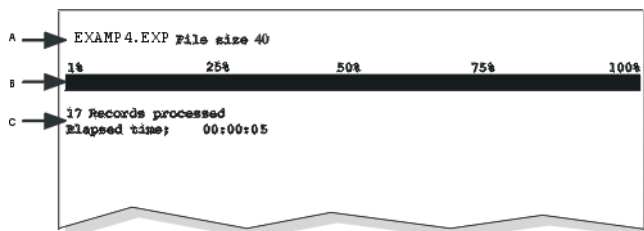
5 Applications Found			
EXAMP1EXP	35	12.06.97	15:31
EXAMP2EXP	40	12.06.97	15:31
EXAMP3.EXP	35	12.06.97	15:31
EXAMP4.EXP	40	12.06.97	15:31
EXAMP5.EXP	40	12.06.97	15:31
Current Directory	[.]	
	[..]	Previous Directory

**NOTE**

DOSDL.EXE recognizes application filenames having a maximum of 8 characters.

It is advised to select filenames that respect this limit before exporting them from EasyGen™.

Choose one application program and press <ENTER>. The screen displays the following:



Key:

- A) File name and size
- B) Loading status bar
- C) Reply message

To interrupt the installation of the application at any time, just press <CTRL>+<BREAK> via the computer keyboard.

It is also possible to load the application using DOSDL.EXE directly from the DOS prompt. Enter the command:

DOSDL /f filename.EXP

if the COM1 serial port is used

DOSDL /L2 /f filename.EXP

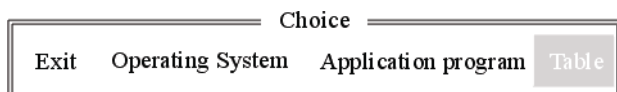
if the COM2 serial port is used.

B1.3. DOSDL.EXE - Tables

To load the tables relative to the application loaded on the Formula terminal, the item "File – Send" can be selected from the EasyGen™ main menu, or the DOSDL.EXE program can be used.

To use this loading possibility it's necessary that you have already carried out the "Export" operation.

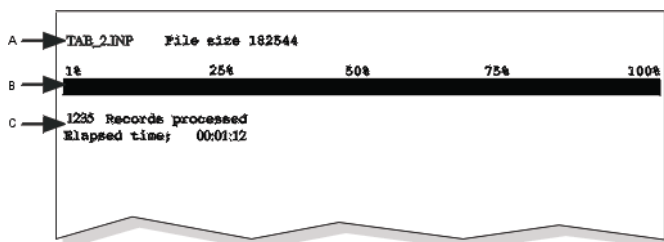
After making the RS-232 connection between the Formula terminal and the host computer and loading the interpreter program onto the Formula terminal, it is necessary to select the "Table" item from the DOSDL "Choice" menu.



A list of all application tables ("INP" extension) present in the current directory appears on the monitor (it is possible to change directories):

		2 Tables found		
		TAB_1.INP	90	12.06.97 15:41
		TAB_2.INP	100	12.06.97 15:41
Current Directory	→	[.]
Previous Directory	→	[..]

Select the desired table and press <ENTER>; the following appears on the monitor:



Key:

- A) File name and size
- B) Loading status bar
- C) Reply message

To interrupt the installation of a table at any time, just press <CTRL>+<BREAK> on the computer's keyboard.

It is also possible to load the table using DOSDL.EXE directly from the DOS prompt. Enter the command:

DOSDL /f filename.INP

if the COM1 serial port is used

DOSDL /L2 /f filename.INP

if the COM2 serial port is used.

To load the actual table .DAT files to the Formula terminal, it is necessary that they are also present in the same directory. They will then be automatically loaded with the table .INP files.

B.2. ALTERNATIVE FILE LOADING METHOD

Application programs and tables "Exported" (see par. 4.2.8) from EasyGen™ can be sent to the Formula terminal using a serial port under a different operating system than Windows 95/98/2000/ME or NT.

□ Loading an application program

To load the application program in a Formula terminal, in which the interpreter is already loaded, you must use the default protocol with the serial port set to 9600 baud, 8 bits and MARK parity.

The data or command of the "communication frame" (see Appendix C) must be input as a single line [without the terminating characters CR(0D Hex) and LF(0A Hex)], of the file to be sent maintaining the following FIXED sequence:

1. *Appl.CFG* file: contains the configuration parameters of the Formula terminal;
2. *table1.INP* file: contains the first table definition;
3. *table1.DAT* file: this is a "data" table with the format defined in *table1.INP*. This file must be created outside EasyGen™ following the rules described in par. 4.6.1.
The single text "data" line, without the terminating characters CR(0D Hex) and LF(0A Hex) preceded by ESC(1B Hex), "2"(32Hex) and followed by ESC(1BHex), becomes the "communication frame" (see Appendix C) to be sent.
If your application does not require this file go to step 4.
- 2N - 3N. *tableN.INP* Repeat step 2 and 3 for every table defined.
4. *appl.CMD* file: contains the actual application program.

For your convenience "Export" also creates an *appl.EXP* file which is the list without the *table.DAT* files required by your application.

❑ Loading a single table

Similarly, it is possible to send the data of an individual table to the Formula terminal.

This means that a table from an already loaded application program can be changed and the terminal file can be updated without reloading the program.

Using the definition of a table.INP file created with “Export”, you can send the new table.DAT file directly to the Formula terminal using the fixed sequence and format described in points 2) and 3) of the previous paragraph, “Loading an Application Program onto the Formula Terminal”.

Formula®

C COMMUNICATION PROTOCOL

C.1. COMMUNICATIONS FRAME

To receive and/or send data or commands, EasyGen™ uses the following communications frame:



The figure shows the position of each character constituting the communications frame of the information to be transmitted or received.

It is possible to configure each character using the "S" type command (see par. C.4 "Configuring the Terminal (Serial Command "S)").

If these parameters are not defined by the user the default configuration is used.



NOTE

Once a value is attributed to a particular character in the communications frame, it is important not to attribute it to another character from the same frame.

In case more than one terminal is connected to the same host computer, it is essential to configure the Station Address character so that it assumes different values for each terminal.

The Formula terminal can use two different Block check characters (BCC). These can be set by using the serial command "S" described in Appendix C:

- LRCC = Longitudinal Redundancy Check Character
- Checksum 256

The use of any type or other is completely arbitrary. The only existing limitation regards the specifications of the management program residing in the host computer. Depending on the BBC used, the calculating method of the two characters that compose it varies:

- If LRCC is used, it is necessary to calculate the xor operation of all characters preceding the LRCC (including any Start of Block and <etb> or <etx> characters). The ASCII character corresponding to the value thus obtained must be inserted in the communications frame.
- If Checksum 256 is used, it will be necessary to calculate the sum of all the characters preceding the Block Check Characters (including any "Start of Block" and <etb> or <etx> characters) and subtract from the result 256 dec.

The value obtained must be broken down into the two nibbles that its composed of (groups of 4 bits) and the two ASCII characters corresponding to the values obtained must be inserted in the communications frame.

EXAMPLE

Let us imagine that you have to send the command <ESC>5<ESC><CR> using BBC = LRCC. The value of the Block Check Characters must be calculated as follows:

ASCII		Hexadecimal		Binary	
ESC	=	1B ₁₆	=	00011011	xor
"5"	=	35 ₁₆	=	00110101	xor
<ESC>	=	1B ₁₆	=	00011011	xor
<CR>	=	0D ₁₆	=	00001101	=
"8"		38 ₁₆	=	00111000	

If you carry out the xor operation with hexadecimal values, the result will be 38₁₆.

If you carry out the xor operation with binary values, the result will be 00111000.

The ASCII character "8" must be inserted into the communication frame.

EXAMPLE

Let us imagine that you have to send the message <ESC>7CKSUM<ESC><CR> using BCC=Checksum module 256. The Block Check Characters value must be calculated as follows:

ASCII		Hexadecimal		Binary	
ESC	=	1B ₁₆	=	00011011	+
"7"	=	37 ₁₆	=	00110111	+
"C"	=	43 ₁₆	=	01000011	+
"K"	=	4B ₁₆	=	01001011	+
"S"	=	53 ₁₆	=	01010011	+
"U"	=	55 ₁₆	=	01010101	+
"M"	=	4D ₁₆	=	01001101	+
<ESC>	=	1B ₁₆	=	00011011	+
<CR>	=	0D ₁₆	=	00001101	=
		1FD ₁₆			11111101

If you calculate the sum with hexadecimal values, you obtain the result 1FD₁₆; divide it by 100₁₆ (256Dec) and you are left with FD₁₆. In practice only the last two less important digits of the hexadecimal sum are considered.

If you calculate the sum with binary values, you obtain 1.1111.1101; divide it by 1.0000.0000 Bin. (256Dec) and you are left with: 1111.1101, when that is changed into hexadecimal form it is FD₁₆. In practice only the last two less important nibbles of the binary sum are considered.

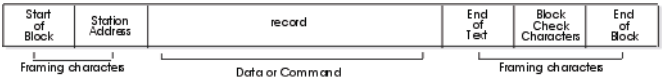
The value obtained must be broken down into the two nibbles that make it up, thus the characters to be inserted into the communication frame are the ASCII values "F" and "D".

C.2 DATA TRANSMISSION

The transmission of data from the Formula terminal towards the host computer depends on the interface used.

C2.1 RS-232 and RS-485 Interfaces

Data transmission for both of these interfaces is virtually identical. To receive and/or send data in this mode every unit of information must have the following structure:



C.2.2 Eavesdrop Connection Interface

In this operating mode the Formula terminal will begin to transmit data contained in the table to the host computer as soon as it is inserted into the cradle. This data is not sent record by record but rather field by field. To do this, each field is changed into a record before being transmitted. While this is being done the separators are removed.

Let us imagine that the following record has been input to the terminal:

0	5	/	0	3	/	9	0		1	5	:	4	5		8	8	0	7	4	3	2	9		0	2	
---	---	---	---	---	---	---	---	--	---	---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	---	--

Before being sent to the host computer it is broken down as shown below:

0	5	/	0	3	/	9	0	1	5	:	4	5	8	8	0	7	4	3	2	9	0	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



ATTENTION

This operating mode is not available for the F732 terminal.

C.3 HOST COMPUTER COMMANDS

Using a serial line you can handle the data stored in a Formula terminal directly from the host computer.

The data handling commands are sent to the Formula terminal as groups of characters having the character <ESC> (ASCII 1B₁₆) at the beginning and end. The general command format is the following:



Command type: represents the instruction to perform and can take the values described below:

0	Download data
1	Delete
2	Send record
5	Enquiry
8	Terminal information
B	Start application
F	File field structure
R	File structure
S	Serial programming
Z	Clock tasks

Command data: this command data takes alphanumeric values according to the instruction that you send to the terminal.

0 - Download data

Use this command when you want to download a data file or program file.
Command data can take the following values:

- filename* request all records of specified file, if it exists. The *filename* must always consist of 8 characters;
- ** request all records of the existing files;
- +* request download of current record, record pointer is not updated;
- +nnn* request download of *nnn* records beginning with the current one. The record pointer is not updated;
- #* request zero-setting of downloaded indices;



ATTENTION

During data download not more than three empty consecutive records must be present, otherwise the operation in progress end.

The Formula terminal transmits the *n* data records:

- 1)

F	i	l	e	n	a	m	e
---	---	---	---	---	---	---	---

 HEADING
- 2)

RECORDS
- $n-2$)
- $n-1$)

<	E	O	F	>
---	---	---	---	---

 END OF FILE
- n)

<	E	O	T	>
---	---	---	---	---

 END OF TRANSMISSION

If an individual record is requested, the Formula terminal sends the record to be transmitted and the message <EOT> to the host computer:

- RECORD
- | | | | | |
|---|---|---|---|---|
| < | E | O | T | > |
|---|---|---|---|---|

 END OF TRANSMISSION

1 - Delete

This command deletes/clears one or more data or program files.
Command data can take the following values:

<i>filename</i>	the specified file is deleted, whether it is a data file or a program file;
*	all output file records are deleted;
-	deletes the current record and moves the pointer by one position.

The answers are:

"DEL"	Records deleted.
"CLR"	Records deleted.
"WAI"	Wait, deleting.
"FNF"	File not found.

2 - Load record

Use this command when you want to load a record into the Formula terminal.
Command data takes the value of the record as its value.

The answers are:

"ACK"	Records saved.
"NAK"	Record not conforming with previously given definition or file full.

5 - Enquiry

Each time that the Formula terminal receives this command, it transmits a message that defines its internal status to the host computer.
Command data does not take up any value.

The answers are:

"RUN"	Terminal connected.
-------	---------------------

8 - Terminal information

This command allows the host computer to identify the type of Formula terminal.
Command data can take the following values:

no value	Formula terminal transmits its name to the host computer.
\$	Formula terminal sends version and firmware release to the host computer in the format v.x/r.y;
*	Formula terminal sends the RAM extension in the format XXX Kbyte.

The answers are:

name	name of the terminal.
v.X / r.Y	X is the version and Y is the release of the terminal in use.
XXX	RAM memory of the terminal.
"NAK"	Wrong command.

B - Start application

This command allows the host computer to send the application program start command.
Command data does not take up any value.

F – File fields structure

This command defines the structure of the file fields. It is not used directly by the user but it can be found inside the files that are obtained by exporting the application in use in EasyGen™.

R – File structure

This command defines the file structure. It is not used directly by the user but it can be found inside the files that are obtained by exporting the application in use in EasyGen™.

S - Serial programming

This command allows you to configure the Formula terminal from the host computer via serial line. Due to the complexity of the topic, refer to the par. C.4 "Configuring the terminal (serial command "S")".

Z - Clock tasks

This command allows the host computer to carry out special tasks on the internal clock of the Formula terminal.

Command data can take the following values:

YYMMDDHHMMSS	date and time setting;
?	request date and time of Formula terminal.

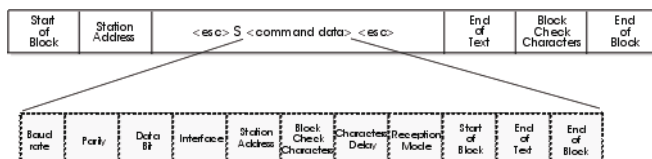
The answers are:

"ACK"	Operation ready.
"NAK"	Wrong format.
YYMMDDHHMMSS	Year, Month, Day, Hours, Minutes, Seconds.

C.4 CONFIGURING THE TERMINAL (SERIAL COMMAND “S”)

Configure Formula terminal communication parameters by choosing the items Tools - Options from the main menu of the EasyGen™. You can edit the configuration by using the serial command “S”.

Command data must have the following structure:



The parameters that must not be edited, must in any case be included in the command data with the value unchanged.



NOTE

In the following tables the default values are placed in a square.

☐ Baud rate

VALUE	SERIAL COMMAND
300	0 = 30 ₁₆
600	1 = 31 ₁₆
1200	2 = 32 ₁₆
2400	3 = 33 ₁₆
4800	4 = 34 ₁₆
9600	5 = 35 ₁₆
19200	6 = 36 ₁₆

This parameter allows you to set the transmission speed.

Parity

VALUE	SERIAL COMMAND
Odd	0 = 30 ₁₆
Even	1 = 31 ₁₆
Mark	2 = 32 ₁₆
Space	3 = 33 ₁₆
None	4 = 34 ₁₆

Allows you to set parity check

☐ **Data Bit**

VALUE	SERIAL COMMAND
7	0 = 30 ₁₆
8	1 = 31 ₁₆

The instrument uses ASCII code exclusively, so 7 bits are sufficient for communication.

☐ **Interface**

VALUE	SERIAL COMMAND
RS-485	0 = 30 ₁₆
RS-232	1 = 31 ₁₆
EAVESDROP	2 = 32 ₁₆

Allows you to define serial interface for connection between Formula terminal and host computer.

☐ **Station Address**

This parameter defines the Formula terminal address unequivocally. The Station Address must be defined as a two-digit number from 00 to 99. The default value of this parameter is "01".

❑ Block Check Characters

VALUE	SERIAL COMMAND
LRCC	0 = 30 ₁₆
Checksum 256	1 = 31 ₁₆
Disable	none = 20 ₁₆

When this parameter is enabled all data transmitted to host computer or Formula terminal is completed by Block Check Characters that carry out a transmission check. If transmission is successful the receiving instrument sends an ACK character. By contrast if transmission occurs with a wrong checksum, the receiving instrument replies with a NAK character.

It is advisable to always define the Block Check Characters when connecting host computer to Formula terminal because this is the only way to check that data is not lost or corrupted during transmission (refer to the paragraph entitled "Communications frame").

❑ Characters Delay

This parameter defines the time the Formula terminal waits before transmitting the next character.

This delay must be defined as a one-digit number from 0 to 9 and is calculated as:

$$\frac{\text{Delay sec.}}{50} = (\text{command data}) \text{ msec.}$$

For example, set *command data* to "9" to obtain a Delay of 0.45 seconds:

$$\frac{0,45 \text{ sec.}}{50} = 9 \text{ msec.}$$

☐ Reception Mode

VALUE	SERIAL COMMAND
Half Duplex	0 = 30 ₁₆
Full Duplex	1 = 31 ₁₆

This parameter allows you to set the type of communication.

In Full Duplex mode the Formula terminal can transmit and receive data simultaneously; in Half Duplex mode the Formula terminal cannot receive data while it is transmitting, so the transmitted data is lost.

☐ Check characters

The block check characters are:

- Start of Block (default <stx>)
- End of Text (default <etx>)
- End of Block (default <cr>)

The next section provides all the possible values associated with block check characters.

You cannot associate the same value to different block check characters.

soh	01 ₁₆
stx	02 ₁₆
etx	03 ₁₆
eot	04 ₁₆
enq	05 ₁₆
ack	06 ₁₆
bel	07 ₁₆
bs	08 ₁₆
ht	09 ₁₆
lf	0A ₁₆
vt	0B ₁₆
ff	0C ₁₆
cr	0D ₁₆
so	0E ₁₆
si	0F ₁₆
dle	10 ₁₆

dc1	11 ₁₆
dc2	12 ₁₆
dc3	13 ₁₆
dc4	14 ₁₆
nak	15 ₁₆
syn	16 ₁₆
etb	17 ₁₆
can	18 ₁₆
em	19 ₁₆
sub	1A ₁₆
esc	1B ₁₆
fs	1C ₁₆
gs	1D ₁₆
rs	1E ₁₆
us	1F ₁₆
none	20 ₁₆