PL7 Micro/Junior/Pro

Brief presentation of PL7

eng V4.0

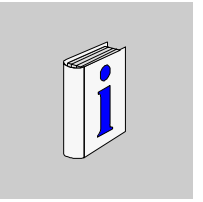
35002981 00

Related Documentation

Document set All documents relating to the installation of PL7 software can be found on the documentation CD-ROM supplied with the software.

Table of Contents

About the book



At a Glance

Document Scope

The aim of this brief presentation of PL7 is to provide a concise overview of the software.

Validity Note

This document takes into account the changes implemented in PL7 V4.

Related Documents

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

General

1

Introduction

Aim of this Chapter

This chapter introduces the PL7 software user interface.

What's in this Chapter?

This Chapter contains the following Maps:

General characteristics of the user interface

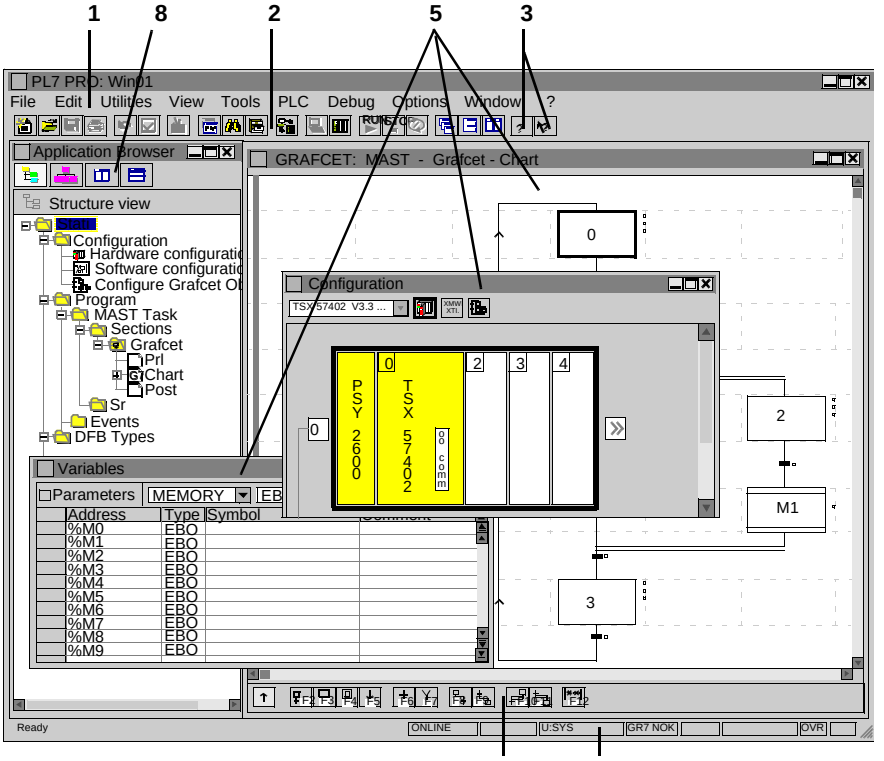
At a Glance

The packages use all the standard Windows functions:

- mouse or keyboard,
- drop-down menus,
- browsers,
- palettes and toolbars with icons,
- several tools for the same function,
- on-line Help and information balloons.

Illustration

The PL7 screen shown below provides an example of the numerous tools available:



Elements and functions

This table describes the different elements that make up the PL7 screen:

Number	Element	Function
1	Menu bar	Allows access to all the software functions,
2	Toolbar	Allows quick access by mouse to all basic functions,
3	Help	Provides information about the software,
4	Browsers	Allows direct access to different editors
5	Editors	Allows creation, debugging and operation of applications,
6	Graphics palette	Allows direct access to current editor tools,
7	Status bar	Shows a range of information associated with the software.

The PL7 toolbar

Presentation

The software's basic functions can be accessed quickly via the toolbar, using the mouse.
Access to the different functions is dynamic and varies according to the context.

Illustration

The PL7 toolbar is displayed as follows:

Elements and functions

This table gives the function of each element in the toolbar:

Element	Function	Element	Function
	New application		Local mode
	Open an application		Online mode
	Save the application	RUN	PLC changes to RUN
	Print all or part of the application	STOP	PLC changes to STOP
	Undo last modifications		Start / Stop the animation
	Confirm modifications		Organize windows so that they overlap
	Go to		Tile windows horizontally
	Application browser		Tile windows vertically
	Cross references		Help
	Function library		What's this?
	PLC <-> terminal transfer		

Note: All these functions can also be accessed via the menu.

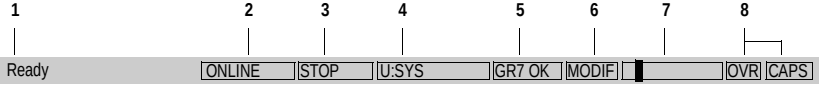
The PL7 status bar

At a Glance

The status bar, situated at the bottom of the screen, shows a range of information associated with operational aspects of the software.

Illustration

The PL7 status bar appears thus:



Elements and functions

This table describes the different zones that make up the status bar:

Number	Zone	Function
1	Information	supplies information concerning menu commands, toolbar icons and the different editors when these are selected.
2	Operating mode	indicates the current operating mode (offline, online).
3	PLC state	indicates the PLC state (Run, Stop, faulty, etc.).
4	Network address	gives the network address of the PLC.
5	Grafcet mode	indicates whether Grafcet mode is used in the application.
6	Modification in progress	indicates that the current application has not been saved or is different from the back-up.
7	Animation indicator	indicates that the PLC is in online mode.
8	Keyboard functions	indicates the status of the **Insert** and **All Caps** keyboard functions.

PL7 on-line Help

At a Glance

The PL7 on-line Help describes the implementation of different editors within the software. It also provides a wealth of information about:

- users (access rights),
- general information about PL7 (application structure, adressing bit and word objects, memory management, etc.)
- PL7 language instructions (functions, syntax, operands),
- using PL7 (programming, debug, diagnostics),
- TSX Micro and Premium tasks (Regulation, Counting, Weighing, etc.).

Access mode using PL7

Two access modes are proposed:

- from the **(Help topics)** (See *Help Topics Browser, p. 15*) browser,
- directly from a PL7 screen **(Contextual Help)** (See *PL7 contextual Help, p. 17*).

Help Topics Browser

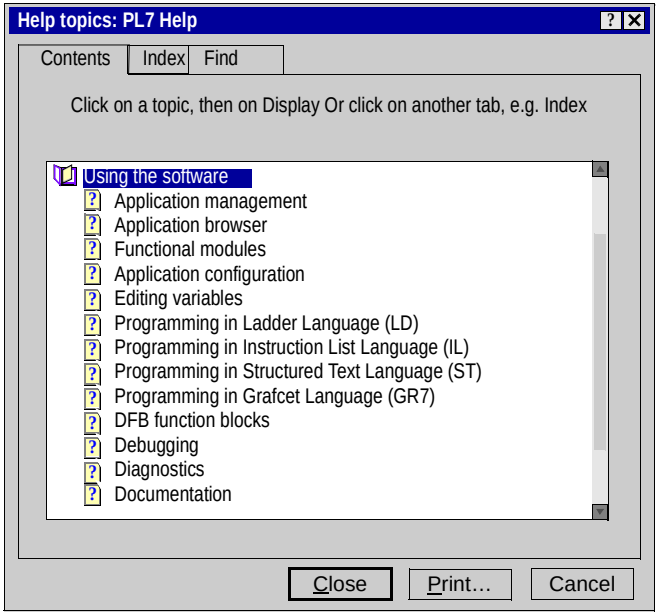
At a Glance

The **Help Topics** browser provides for three types of search:

- from the **Contents**, which displays a view of the all the different chapters of the Help system,
- using the **Index**, which displays an alphabetical list of key words,
- using the **Find** mode, which displays all the words used in the on-line Help in alphabetical order.

Illustration of the browser

The following illustration shows the browser open at **Contents**



Accessing the browser

Contents tab

Step	Action
1	Select the **Index** command from the **?** menu or click on the icon ?.
2	Select then open the required directory.

Index tab

Step	Action
1	Select the **Find ...** command from the **?** menu or click on the  icon then select the **Index** tab.
2	Enter the key word.
3	Select then open the required topic.

Find tab

Step	Action
1	Click on the  icon then select the **Find** tab.
2	Enter the word to be found.
3	Select then open the required topic.

PL7 contextual Help

At a Glance

Contextual Help is used to directly access information from the selected element.

Accessing the Contextual Help

Two exclusive modes of access are used to access Contextual Help.

Standard screens

Step	Action
1	Select the **What's this?** command from the **?** menu or click on the  icon,
2	Select the element for which you require technical information (menu, screen, toolbar, etc.).

Modal dialog boxes

Step	Action
1	Click on the  icon of the current element.

The application browser

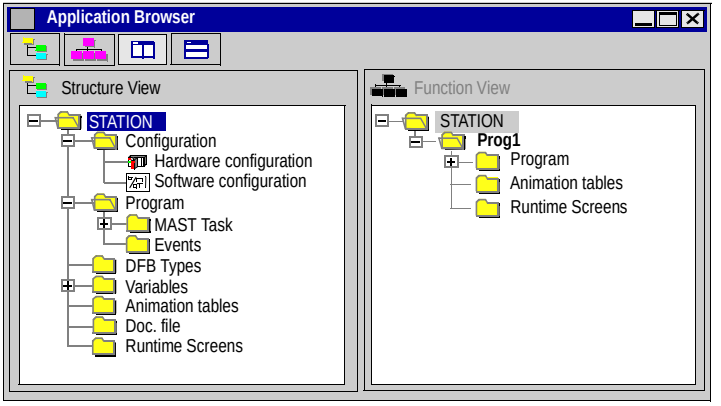
At a Glance

The application browser displays the contents of a PL7 application in tree diagram form.
Two types of representation are offered:

- structural: standard view corresponding to the PLC's processing order,
- functional: view of the application in functional modules corresponding to the functions of the automatic operation.

Illustration

The following screens show the two modes of representation.



Accessing the browser

The following table shows the different ways of accessing the application browser.

From:	Action
the contents	Select the **Tools** → **Application Browser** command.
the tool bar	See *The PL7 toolbar, p. 12*

Note: By default, the application browser opens a partial display of the directory tree.

- + in front of a directory indicates that it can be opened,
- - in front of a directory indicates it can be closed.

To open or close a directory, click on + or -, or use the left or right arrows on the keyboard.

Which software for which mode

The following table shows the types of representation available for each software package:

	PL7 Micro	PL7 Junior	PL7 Pro
Structure view	yes	yes	yes
Function view	no	no	yes

Accessing different views

Click on the following icons to access the required display mode:

Icon	Action
	displays structure view
	displays function view
	juxtaposes structure view and function view
	superimposes structure view and function view

Structural representation

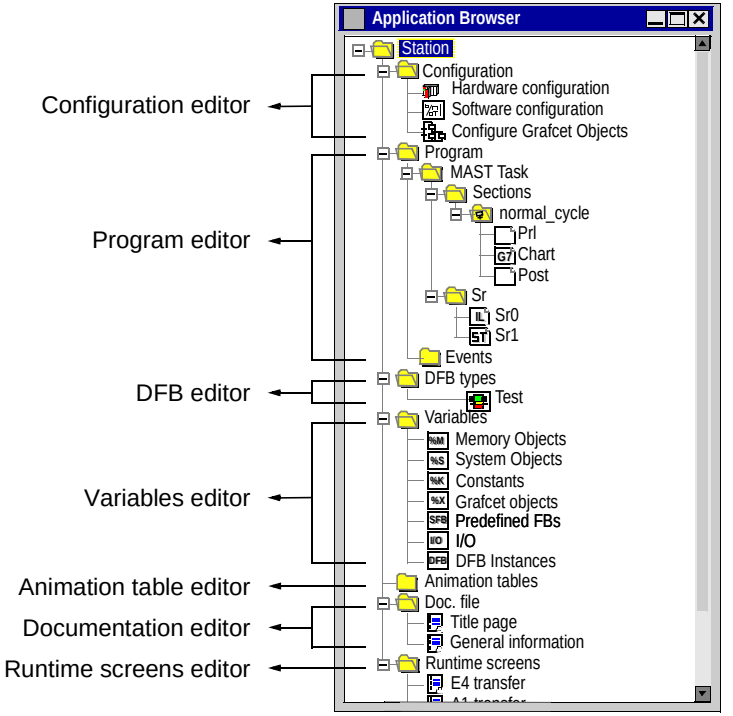
At a Glance

Structural representation shows the contents of an application as a directory tree. It is used to move around inside an application and provides direct access:

- to the hardware, software and Grafcet configuration,
- to the programs,
- to the DFBs contained within the application,
- to the data,
- to the animation tables,
- to parts of the file (general information, title page),
- to the runtime screens.

Illustration

The following screen shows the tree structure of an application.



Functional representation

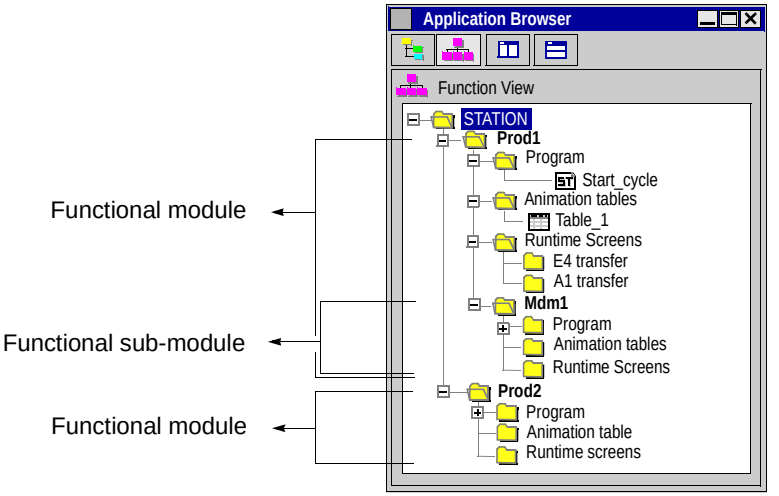
At a Glance

The functional representation shows the application broken down into **functional modules** corresponding to the different functions of the application's automatic processes.
A functional module is a grouping of program elements (sections, macro-steps, animation tables, runtime screens, etc.).

Note: This function is available with PL7 Pro software.

Illustration

The following screen shows an application broken down into function modules.



Introduction to PL7 editors

2

At a Glance

Overview

This chapter introduces the different editors offered by PL7.

Note: For further information (functions, access, etc.), refer to either of the following:
- PL7 on-line Help,
- The various manuals available on CD-ROM.

What's in this Chapter?

This Chapter contains the following Maps:

The configuration editor

At a Glance

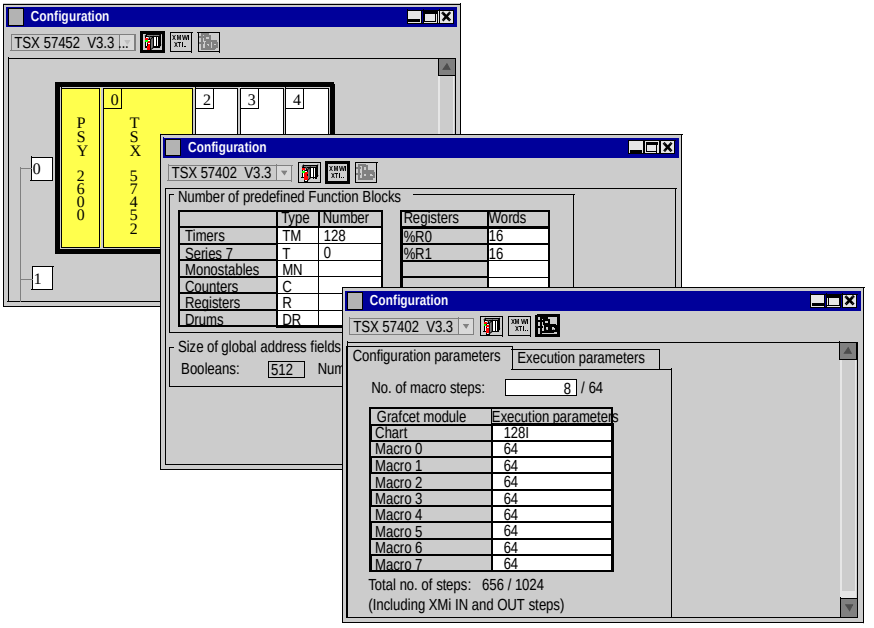
The PL7 configuration editor maintains the following functions for each application:

- hardware configuration
- software configuration,
- Grafcet configuration, wherever programming is in Grafcet.

When connected, the configuration editor also maintains the debugging, adjustment and diagnostic functions.

Illustration

The following screens show the different views of the configuration editor.



How to access the editor

The following table shows the different ways of accessing the application browser.

From:	Action
the menu bar	Select the **Tools** → **Configuration** command.
the application browser	Double click on the required configuration type, or select using the arrow keys and confirm with **Enter**.

How to change from one configuration type to another

Click on the following icons to change from one configuration type to another. :

Icon	Action
	displays the hardware configuration,
	displays the software configuration,
	displays the Grafcet configuration,

Hardware Configuration

The configuration editor is a user-friendly graphic interface, for defining and configuring the different parts of the PLC:

- rack,
- supply,
- processor,
- task modules,

Software configuration

The configuration editor maintains application software parametering by reporting:

- the number of function blocks,
- the number of registers,
- the size of global address fields.

Grafcet objects configuration

The configuration editor is used to define Grafcet objects (steps, macro-steps, etc.), and set operating parameters (number of active steps and transitions).

Program editors: General

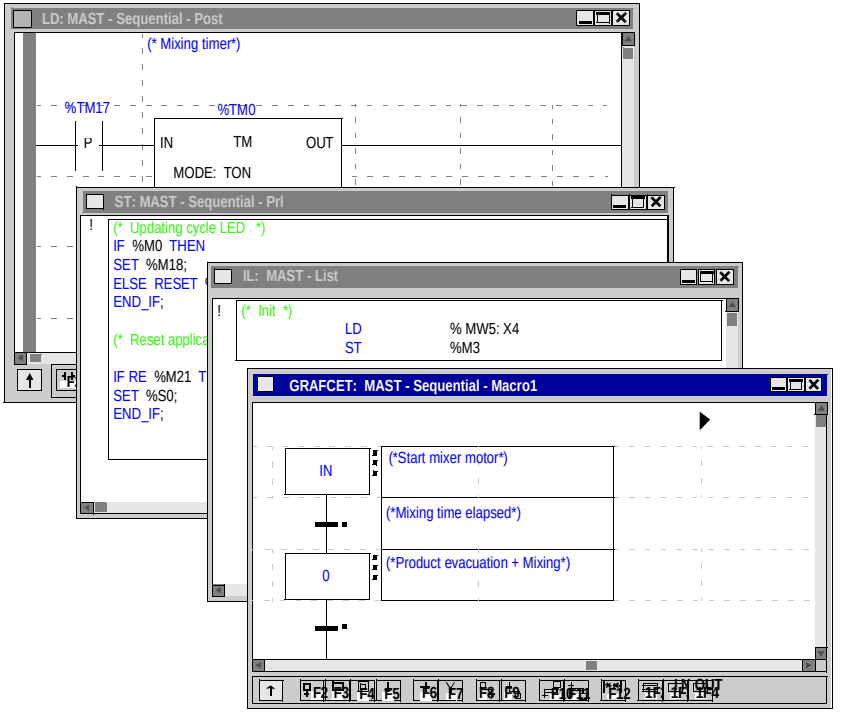
At a Glance

The program editors are used to program functions and tasks implemented by the application.
Four program editors are offered:

- ladder language editor (LD),
- instruction list language editor (IL),
- structured text language editor (ST),
- Grafcet language editor (G7)

Illustration

The following illustration shows a view of all the different program editors.



How to access an editor

The following table shows the procedure for accessing a program editor.

If the application	From the application browser:
does not have a program	Create or import a program module (section, subroutine, event, DFB) in one of the proposed languages. To do this, refer to the Operate Modes on-line help.
has one or more program module(s)	Open the **Program** directory, select the required program module then double-click on it, or select it using the arrow keys and confirm with **Enter**.

Program editor: Ladder Language (LD)

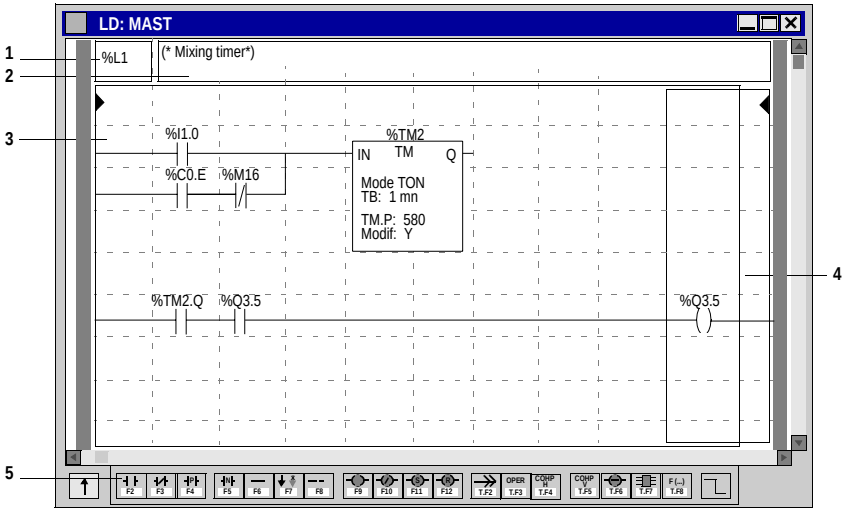
At a Glance

The Ladder Language editor is a graphics editor which is used to construct ladders (transcription of relay diagrams).
This editor is structured in zones, with tools and functions which can be directly accessed using the mouse or the keyboard. They include:

- basic tools (rungs, Boolean wires, spools, operation blocks, etc.),
- immediate call-up of tools for the assisted entry of library functions:
- direct access to a subroutine from a call program,
- different display modes.

Illustration

The following screen shows the different zones which make up the Ladder Language editor.



Elements and functions

The following table shows in brief the different elements that make up the editor.

Number	Element	Function
1	Label zone	is used to enter a label
2	Comment zone	is used to enter a comment
3	Test zone	is used to locate elements such as rungs, function blocks, etc.
4	Action zone	is used to locate elements such as spools and operation blocks.
5	Graphics palette	is used to access different graphic language symbols directly.

Entry modes

The software suggests two entry modes:
- with comment (default entry mode),
- without comment.

This last mode allows graphic elements to be entered without supplying input information, and batches this task when the ladder is completed.

Display modes

Different display parameters allow the display mode to be adapted to the user's requirements.
These different parameters are:
- normal view (default entry mode),
- collapsed view which displays a greater number of ladders while retaining the same level of information.
- display of operands as an address, a symbol, or both simultaneously,
- display of symbols limited to 10 characters (short text) or shown in full (long text: maximum of 32 characters).

How to change modes

The following table shows the procedure for changing modes.

Mode	Command	Keyboard shortcut
Entry	**Edit** → **Enter with Comment** is used to switch from one mode to another.	-
Display	**View** → **Collapsed, Normal**. **View** → **Addresses**. **View** → **Symbols**. **View** → **Symbols & Addresses**. **View** → **Short Text, Long Text**.	- Ctrl + E Ctrl + F Ctrl + H -

Moving around the editor

The following table shows the procedure for moving around the editor using the keyboard.

Movement	Key (s)
From cell to cell	Arrow
To the first column of the rung	Home
To the last column	End
To the next page	PgDn
To the previous page	PgUp
To the start of the ladder	Ctrl + Home
To the end of the ladder	Ctrl + End

Program editor: Instruction List (IL) Language

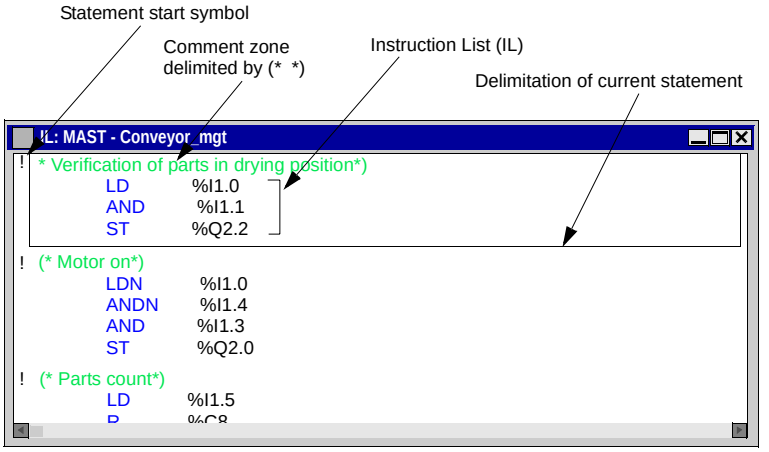
At a Glance

The Instruction List language editor is used to write logic and digital processing in Boolean form.
This editor uses functions such as:

- automatic imposition (alignment of instructions and operands),
- entry and display of operands in symbol and/or address form.
- Assisted entry:
  - function block instructions (%TMi, %Ci, etc.),
  - library functions,
- color display of the key language words and comments.

Illustration

The following screen shows an example of a program constructed in Instruction List language.



Moving around the editor

The following table shows the procedure for moving around the editor using the keyboard.

Movement	Key (s)
From character to character	Arrow keys
From word to word	Ctrl + left and right arrow keys
To the beginning of the line	Home
To the end of the line	End
To the start of the program	Ctrl + Home

Movement	Key (s)
To the end of the program	Ctrl + End
To the previous page	PgUp
To the next page	PgDn

Program editor: Structured Text language (ST)

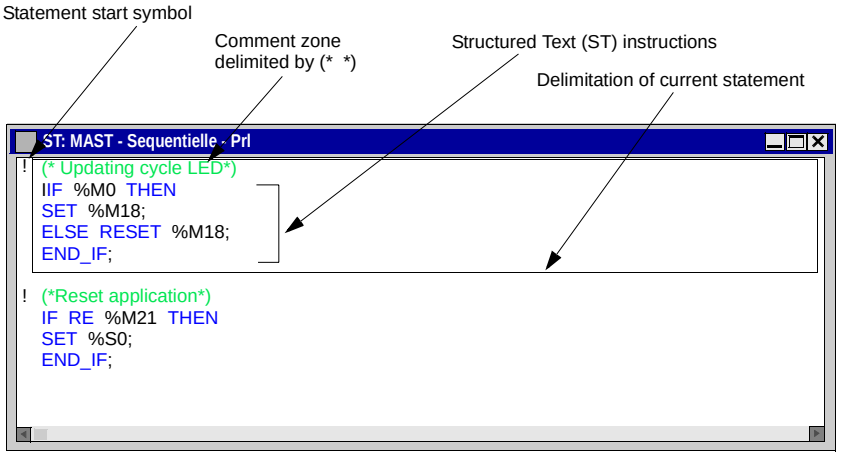
At a Glance

The Structured Text language editor is used to write logic and digital processing in a structured form (data processing type).
This editor uses functions such as:

- entry and display of operands in symbol and/or address form.
- assisted entry of library functions,
- color display of the key language words and comments.

Illustration

The following screen shows an example of a program constructed in Structured Text language.



Moving around the editor

The following table shows the procedure for moving around the editor using the keyboard.

Movement	Key (s)
From character to character	Arrow keys
From word to word	Ctrl + left and right arrow keys
To the start of the line	Home
To the end of the line	End
To the start of the program	Ctrl + Home
To the end of the program	Ctrl + End
To the previous page	PgUp
To the next page	PgDn

Program editor: Grafcet language

At a Glance

The Grafcet editor is used to represent the functioning of a sequential operation in a structured and graphic form.

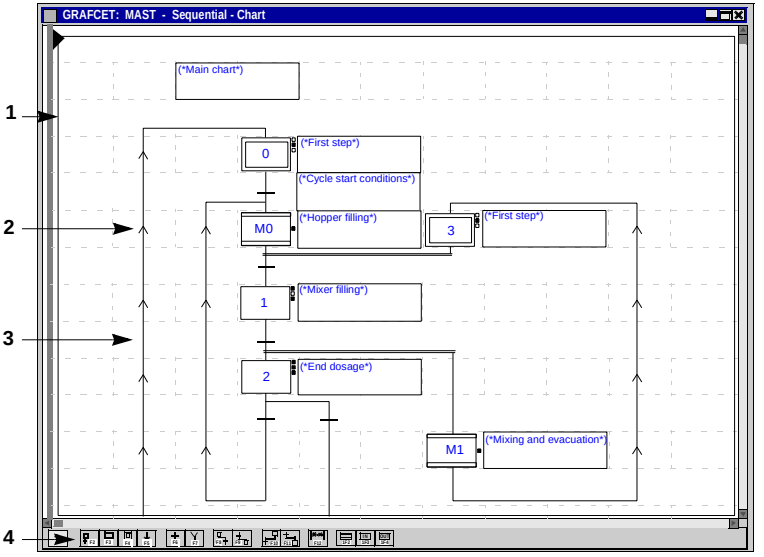
This editor is made up of 8 pages of 14 lines and 11 columns that define those cells that are each able to receive a graphic element.

It has numerous tools that allow user-friendly entry such as:

- a graphics palette directly accessible using the mouse or keyboard (steps, transitions, links, cross-references, macro-steps, etc.),
- direct access to programming of actions or transition conditions,
- automatic numbering of steps,
- a display using Grafcet pages with step and transition lines,
- a simplified entry of comments,
- two display modes.

Illustration

The following screen shows the different zones which make up the Grafcet Language editor.



Elements and functions

The following table shows in brief the different elements that make up the editor.

Number	Element	Function
1	Grafcet Page	allows the graph to be created
2	Step line	allows steps, macro-steps, comments etc. to be entered.
3	Transition line	allows transitions, comments etc. to be entered.
4	Graphics palette	allows different graphic language symbols to be accessed.

Display modes

The software offers two display modes:
- normal view (default entry mode),
- collapsed view.

The latter allows a greater number of Grafcet pages to be displayed while retaining the same level of information.

How to change modes

The following table shows the procedure for changing modes.

Mode	Command
Display	**View** → **Collapsed** or **View** → **Normal**.

Moving around the editor

The following table shows the procedure for moving around the editor using the keyboard.

Movement	Key (s)
From cell to cell	Arrow keys
To the next page	PgDn
To the previous page	PgUp
To the start of the first page	Ctrl + Home
To the end of the eighth page	Ctrl + End

DFB type editors

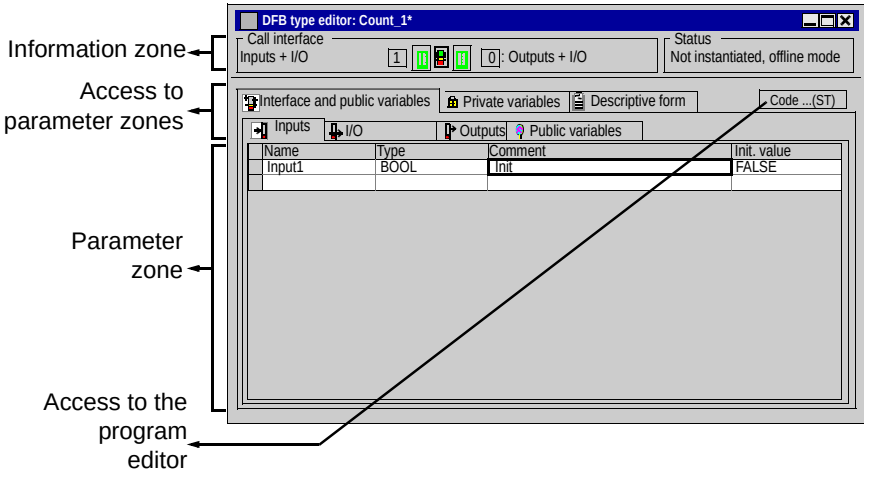
At a Glance

The DFB editor is used for programming user–specific function blocks which work in accordance with the user's application requirements.
These user function blocks are designed for structuring an application. They are used when a programming sequence is repeated within an application, or to freeze a standard programming set.

Note: Creating DFBs requires PL7 Pro. It is possible to use DFBs with PL7 Junior and PL7 Pro.

Illustration

The following screen shows the generic view of the DFB type editor.



How to access the editor

The following table shows the procedure for accessing the DFB type editor.

If the application	Action
does not yet have a DFB type	Create a DFB type. In order to do this, from within the application browser, right-click on the **DFB Types** directory then left-click on **Create**.
has DFB types	Double-click on the required DFB type, or select using the arrow keys and confirm with **Enter**.

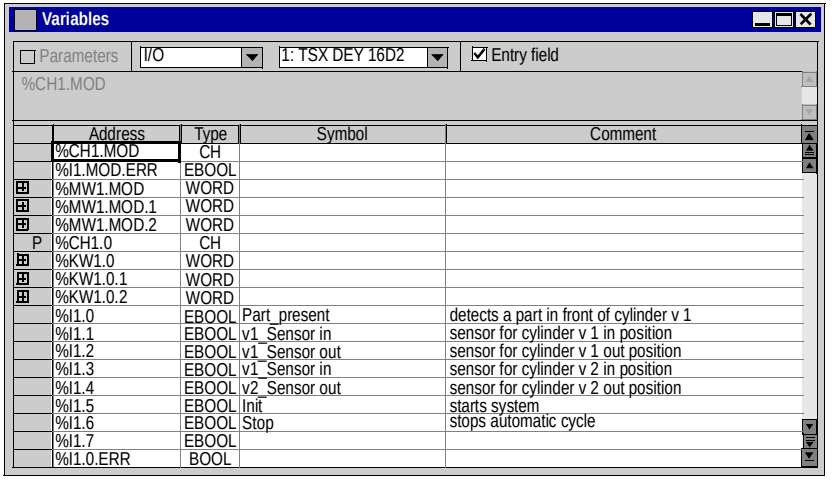
The variables editor

At a Glance

The variables editor is used to:
- create symbols for the different application items (bits, words, function blocks, task modules, etc.)
- create parameters for predefined function blocks (timers, counters, etc.)
- enter constants values and select the display base (decimal, binary, hexadecimal, floating, message),
- instantiate and create parameters for DFB user function blocks.

Illustration

The variables editor appears thus:



How to access the editor

The following table shows the procedure for accessing the variables editor.

Step	Action
1	From the application browser, open the Variables directory.
2	Double-click on the variables type required, or select using the arrow keys and confirm with **Enter**.

Main functions of the editor

Access to the variables is made easier by:
- classification by family and type,
- sort functions (sort by symbols or address),
- the option of displaying all the objects associated with one variable (for example all the bits of a word, all the objects associated with a predefined function block),
- the option to pre-symbolize objects for certain tasks,
- the option to start a wild-card search on a symbol or comment,

- the option to filter the I/O (only displays the input or output variables affecting the process for one module),
- The option to Copy / Paste SFB parameters,
- the option to Cut / Copy / Paste the symbols and comments from a string of variables,
- the option to delete Pre-symbolization,
- bold display of variables used in the program,
- display of overlapping variables used in the program in red,

Animation tables editor

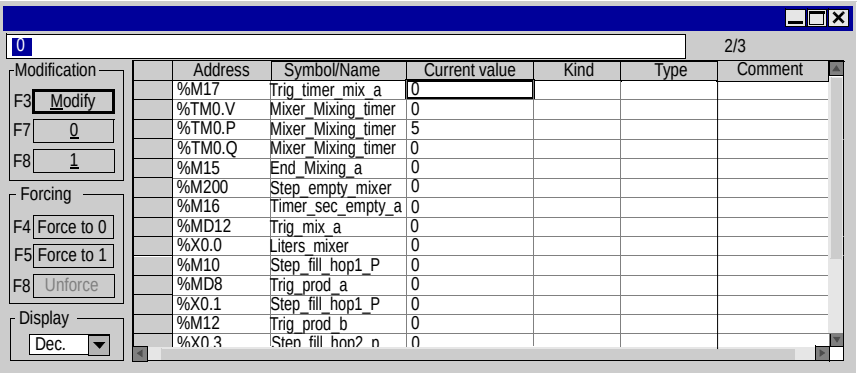
At a Glance

The animation tables editor is used to create tables containing lists of variables to be monitored or modified.
This editor offers functions such as:

- manual creation of tables by entering variables ,
  or
  automatic creation from entire or partial program sections or animated objects in a runtime screen,
- modification of the current value of variables,
- forcing the current value of bit objects,
- choice of the display base for the current value (decimal, binary, hexadecimal, etc.),

Illustration

The animation tables editor appears thus:



How to access the editor

The following table shows the different ways of accessing the animation tables editor.

If the application	From the application browser (manual creation)	From an editor (automatic creation)
does not have an animation table	Create an animation table. In order to do this, from within the application browser, right-click on the **Animation Tables** directory then left-click on **Create**.	Access the program module in which the table has been created, then select the required rung, sequence, instruction or DFB. Select **Initialize Animation Table** (contextual menu). Tables created automatically can be modified later on by deleting or adding new variables.
has one or more animation tables	Open the **Animation tables** directory and select the required table, then double click on it, or select using the arrow keys and confirm with **Enter**.	

Documentation editor

Introduction

The documentation editor is used to configure, view and print the application folder. It is linked with the documentation browser which shows the folder setup in a tree structure.
This editor is used to set :

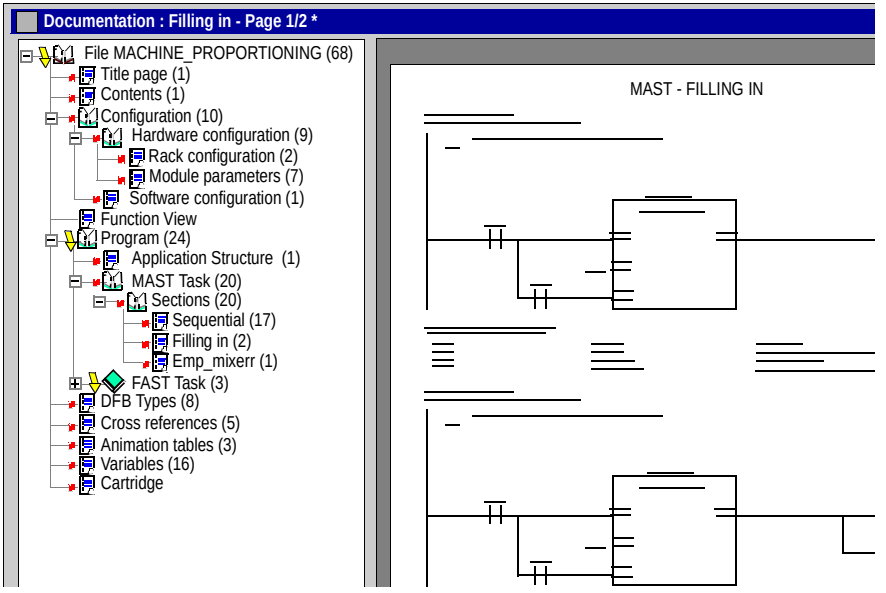
- a title page, containing the designer and project names,
- pages of general information,
- a cartridge.

It automatically generates :

- the contents,
- the application folder : hardware/software and program configurations,
- a list of variables, sorted by address or by symbol,
- cross references, sorted by address or symbol.

Illustration

The following illustration shows a view of the entire documentation editor.



How to access the editor

The following table shows the procedure for accessing the folder editor.

From :	Action
the application browser	Double click on the **Folder** directory or select it using the arrow keys and confirm with **Enter**.

Runtime screens editor

At a Glance

The runtime screens editor is a tool designed to facilitate operation of an automatic process.
Using this tool, a designer can develop process-adapted screens which offer the operator:

- a non-encrypted display of information: explanatory text, dynamic values, color diagrams, etc.
- the possibility to work simply and quickly: dynamic modification and surveillance of PLC variables, etc.

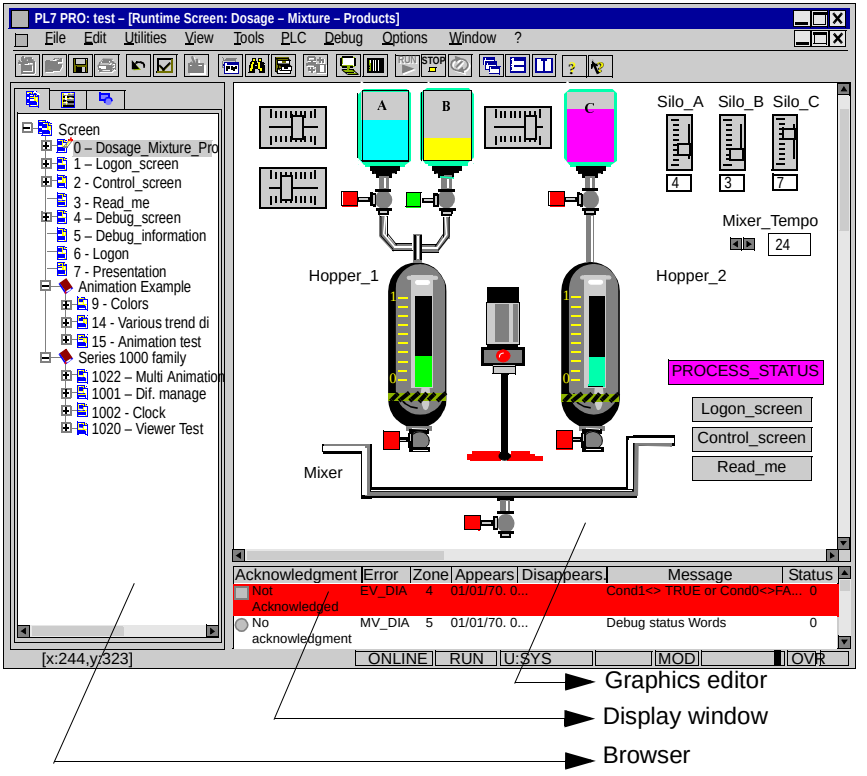
The runtime screens editor has three windows whose sizes can be parameterized:

- the browser used to access the different runtime screens,
- the graphics editor which allows the creation, modification and operation of screen animation,
- the display window, used for help in debugging an application, and providing simple display of diagnostic messages.

Note: Creating or using runtime screens requires PL7 Pro.

Illustration

The following screen shows the different zones which make up the runtime screens editor.



How to access the editor

The following table shows the procedure for accessing the runtime screens editor.

If the application	Action to be taken from the application browser:
does not have a runtime screen	Right-click on the **Runtime Screens** directory then left-click on **Open**
has one or more runtime screens	Open the **Runtime screens** directory and select the required table, then double-click on it, or select using the arrow keys and confirm with **Enter**.

Creating an application

3

At a Glance

Overview

This chapter introduces the principles of creating an application.

Note: For further information (functions, access, etc.), refer to either of the following:
- PL7 on-line Help,
- the various manuals available on CD-ROM.

What's in this Chapter?

This Chapter contains the following Maps:

Creation principles

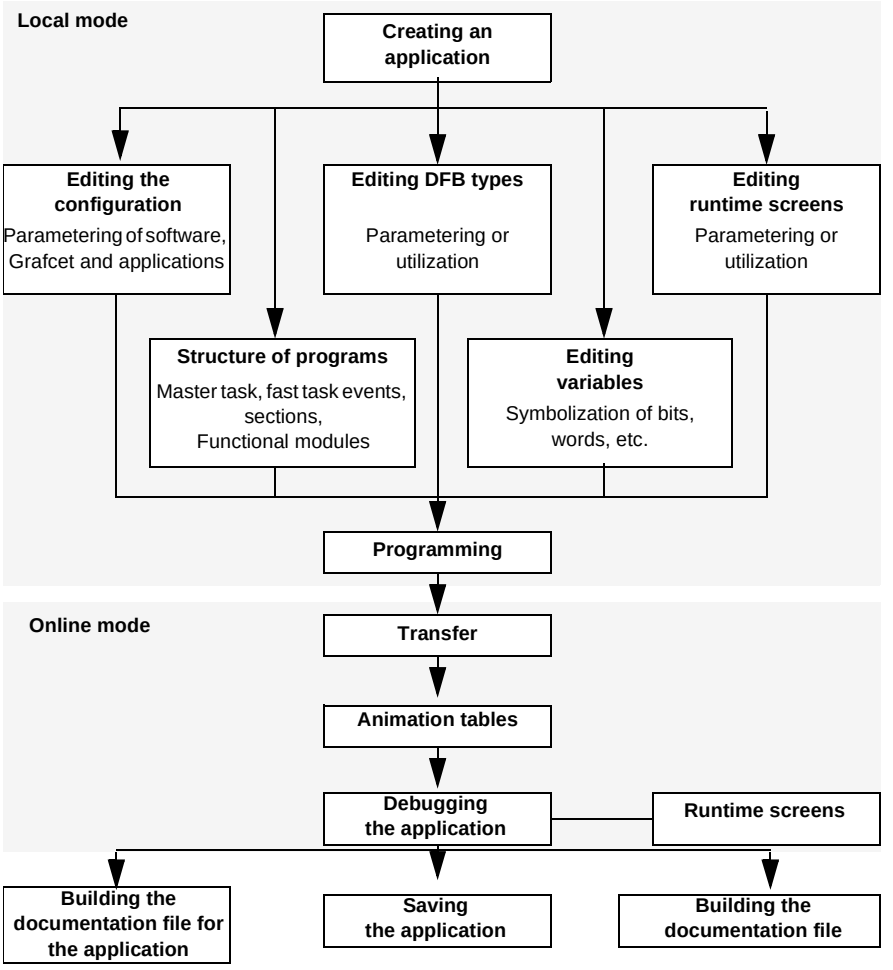
Introduction

2 development stages are involved in creating an application, each one containing several steps:

- a stage in offline mode,
- a stage in online mode.

The different development stages

The flowchart below shows the different stages of developing an application.



Note: Certain development steps require the use of PL7 Junior or PL7 Pro (see PL7 reference manual).

Local mode

Introduction

Local mode (ie. no connection to the PLC) is used for creating or modifying an application at the terminal. This application is located in the work directory on the hard drive.

Online mode

At a Glance

Online mode (ie. connected to the PLC) can be used to modify an application within the PLC.
The following functions can be performed:

- creation/modification of an LD, IL or ST program,
- modification of task period,
- modification of the parameters of predefined function blocks (except size of registers),
- modification of module data and parameters,
- import, application export, PLC in Stop mode,
- addition of predefined function blocks,
- debugging, adjustment,
- creation/modification of runtime screens,
- modification of Grafcet or DFB structure in STOP mode.

Note: When a modification is made in online mode, the application is updated through auto-save to the PLC, and through manual save to the working directory on the hard drive.

Connection problems

This paragraph provides solutions for problems involving connection or disconnection from the PLC. Carry out operations in the recommended order. Only continue to the next operation if the problem is still unresolved.
For a portable terminal:

- Deactivate the energy manager,

For all types of PC:

- Replace the video and mouse drivers with Microsoft drivers.
- Set **Rx Tx** to 0 in the driver parameters.
- Deactivate the anti-virus software (if it is installed).
- Enter 2 as the number of connection attempts in **PLC** → **Define PLC address...** → **Options** → **Retry number**.

Creating an application

Introduction

By creating an application, it is possible to select :
- the type of PLC,
- the processor type,
- the type of memory card.

How to create an application

The table below describes the procedure for creating an application

Step	Action
1	Select the **File** → **New** command.
2	Select the hardware base.
3	Select the processor type. For processors before version V3.0, the Grafcet option must be selected for use within the application (for version > V3.0 processors, a Grafcet application is created by defining a Grafcet section in the master task).
4	Select the appropriate type of memory card. The type of memory card can always be modified later on while configuring the processor.

Program structure

At a Glance

PL7 software allows two types of program structure:

- a mono-task structure consisting of the master task (MAST) structure as a default,
- a multitasking structure which consists of the master task, fast task (FAST) and EVT processing as defaults.

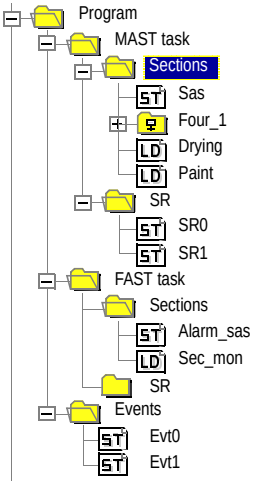
PL7 program tasks are made up of several parts (called sections), and subroutines. Each of these sections can be programmed in a language appropriate to the process to be performed.
This breakdown can be used to create a structured program and to generate or insert program modules with ease.

Task management

Master task and fast task (where programmed) are active by default.
The event task is activated when the associated event occurs.
When a fast task event occurs or its cycle starts, the running of lower priority tasks is stopped in order to deal with the fast task operation. The interrupted task is reinstated when priority task operations are completed.

Example of a multitasking program structure

The figure below gives an example of an application with a multitasking structure.



PL7 program transfer

Role

An entire application can be exchanged with program transfer, apart from the operating screens.
Two transfer directions are possible :

- from the terminal (PC) to the PLC (AP),
- from the PLC to the terminal.

How to transfer an application

PC - > PLC program transfer

Step	Action
1	Select the **AP** → **Transfer Program** command.
2	Select the **PC - > PLC** transfer then confirm with **OK**.
3	If a cartridge for storage of symbols and comments is shown in the configuration, the symbols box may be used : you can choose to complete or delay the symbols transfer to a subsequent transfer run.

PLC - > PC program transfer

Step	Action
1	Select the **AP** → **Transfer Program** command.
2	Select : the **PLC - > PC** transfer then confirm with **OK**.

PL7 data transfer

Role

Data transfer allows variables, symbols and comments concerning the application to be exchanged.
Two transfer directions are possible:

- from the file located on the terminal (PC) to the PLC (AP),
- from the PLC to the file located on the terminal

How to transfer data

Data transfer from PLC -> File

Step	Action
1	Select the **PLC** → **Transfer Data** command.
2	Select the **PLC -> File** transfer.
3	Define the transfer parameters: ● the range of %MW values to be transferred, ● the name of the file where the data is to be stored.
4	Confirm with **OK**.

Data transfer from file -> PLC

Step	Action
1	Select the **PLC** → **Transfer Data** command.
2	Select the **File -> PLC** transfer.
3	Define the name of the file to be transferred.
4	Confirm with **OK**.

PL7 application debugging

At a Glance

PL7 software provides a full set of tools for debugging applications.
A tools palette gives direct access to the main functions:

- a program debug bar which can, for example, be used for:
  - placing of breakpoints,
  - step by step operation of the program,
  - application monitoring.
- a PLC debug screen which offers:
  - information on the status of the application,
  - access to program diagnostics and task modules,
  - access to the display and updating of the realtime clock.
- the Grafcet debug browser which gives a hierarchical view of the chart.
- a Grafcet debug bar which can be used to display or modify chart status.

How to access the debugging tools

The table below shows the access commands for the different debugging tools.

Debugging tool	Access
Program	Select the **Debug** → **Program Debug Bar** command.
PLC	Select the **Debug** → **Access PLC Debug Screen** command.
Grafcet Browser	Select the **Utilities** → **Debug Grafcet** command.
Grafcet Bar	Select the **Debug** → **Grafcet Debug Bar** command.

Note: You must be in online. (See *Online mode, p. 47*)

PL7 Diagnostics

At a Glance

PL7 software can carry out diagnostics in online mode at different application levels such as:

- the **system** (operating system, programming workshop, MMI) using the diagnostics system (PL7 V4 function),
- the **process**, using DFB application diagnostics and DFB command and diagnostics of the operative section,
- the **PLC** (last operation stop , module call stack),
- the **modules** (errors classified according to their category),
- the **program** (cause and source of error),

How to access the diagnostics functions

The PL7 diagnostics functions are allocated to different editors or tools in the PL7 software.
To access the different diagnostics functions, refer to the PL7 on-line Help.

Note: Use of DFBs requires PL7 Junior or PL7 Pro.

Additional tools

4

At a Glance

Overview

This chapter shows the additional tools available in PL7.

Note: For further information (functions, access, etc.), refer to either of the following:
- PL7 on-line Help,
- the various manuals available on CD-ROM.

What's in this Chapter?

This Chapter contains the following Maps:

Converters

Introduction

The converters are designed to portage existing applications simply and effectively to the TSX Micro and TSX Premium PLCs.
Four converters are offered :

- PL7-2 to PL7,
- PL7-3 to PL7,
- ORPHEE to PL7,
- SMC to PL7.

How to access the converters

The following table shows the procedure for accessing the conversion function.

Step	Action
1	Create an empty application to receive the converted application.
2	Select the **File** → **Convert** command.

Note: The SMC converter is an optional converter to PL7 Junior or PL7 Pro (Reference TLX LC SMC PL7 30M).

Import/Export

At a Glance

The Import/Export functions for TSX Micro or TSX Premium PLC applications affect:
- the application, a section, a functional module, an animation table,
- all or part of the IL, LD, ST, Grafcet, DFB Type, Symbol source program modules,

The source file code is in 8 bit ASCII conforming to the standard ISO 8859-1. Code can be entered directly using WINDOWS-compatible editors such as Word in text format (*.TXT).

Note: Binary formats are not accessible (they are encrypted).

File extensions

Files to be imported or exported are of the following types:
- application source marked with *.FEF,
- application source in FNES format, marked with *.FNE,
- functional module source, marked with *.FM,
- LD source, marked with: *.LD,
- IL source, marked with: *.IL,
- ST source marked with: *.ST,
- Grafcet source, marked with: *.GR7,
- symbols source, marked with: *.SCY or *.TXT (compatible with Excel),
- DFB type source, marked with: *.DFB,

How to access the Import/Export function

The following table shows the procedure for accessing the Import/Export function.

Step	Action
1	Select the **File** → **Import/Export** command.

Note: use the **Options** → **Customize** command to identify the access path to source files.

Cross references for an application variable

Introduction

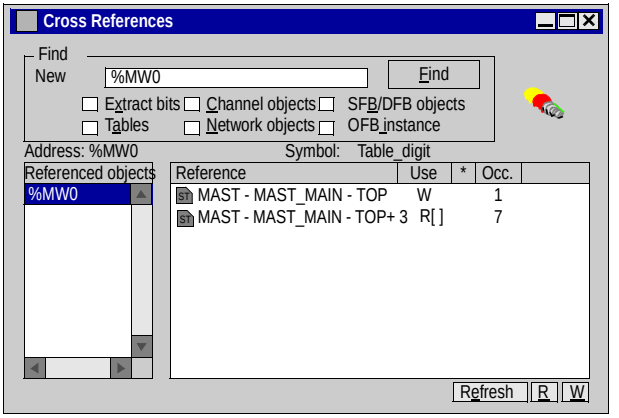
Cross references are mainly used when debugging an application in order to identify the cause of a failed variable.
They are designed to :

- locate all tasks, sections and programming lines where a variable or a DFB type (in address or symbol form) is used,
- go directly the relevant lines,
- display the search and browser archives.

Note: Cross references apply to the application and not to the operating screens.

Illustration

The cross referencing tool appears thus:



How to access the tool

The following table shows the procedure for accessing the cross referencing tool.

Step	Action
1	Select the **Cross references** command from the**Tools** menu.

Replacing a variable in the application

At a Glance

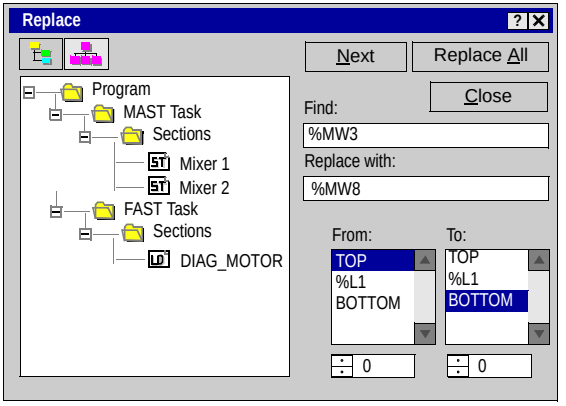
The **Replace Variables** function is used to find and replace an application variable in address or symbol form.
Replacement within the application can be whole or partial, automatic or manual.
It affects the indicated variable as well as related items (eg. word extract bits, etc.) with the exception of activity time for Grafcet step bits (%Xi.T).
Replacement is carried out on the following levels:

- application (in all tasks),
- tasks (Mast, Fast, Evt),
- partial or complete section,
- entire functional modules, (sub-modules included).

Note: The **Replace Variables** function refers to the application and not to the runtime screens

Illustration

The **Replace Variables** tool appears thus:



How to access the tool

The following table shows the procedure for accessing the **Replace Variables** tool.

Step	Action
1	Select the **Replace Variables** command from the **Tools** menu.

Application protection

Introduction

The Protection function for the application is used in local mode for:
- global protection of the application,
- global or partial protection of sections.

Global application protection

This function ensures the read and write protection of an application when it has been transferred within the PLC.
Only the **Run** , **Stop** and **Init** functions are authorized within a protected PLC application.

WARNING

Global protection of an application is irreversible. A protected application cannot be modified. The only option is to load a new application into the PLC.

Failure to observe this precaution can result in severe injury or equipment damage.

Global or partial protection of sections.

For each section, it is possible to set the protection type :
- no protection,
- write protection,
- read and write protection.

How to access the application's Protection function

Application protection

Step	Action
1	Select the **Edit** → **Properties** command.
2	Select the **Protection** tab.

Protecting sections

If protection is	Step	Then …
global	1	Select the **Edit** → **Properties** command.
	2	Select the **Protection** tab.

If protection is	Step	Then …
partial	1	Select the **Programs**,**Task …**, **Sections** directory from the application Browser.
	2	Select the **Protection of included sections** contextual menu (right mouse-click)
individual	1	Select the section to be protected.
	2	Select the **Edit** → **Properties** command.

PL7 access security management

Introduction

PL7 access security management both limits and monitors access to the different PL7 functions.
It can be used to… :

- create/modify a user list,
- import a user list,
- export a user list,
- activate the "PL7 access security management" function,
- change its password.

It is to be applied at the terminal where PL7 software is installed, but not to the application itself

User profiles

Five user profiles are offered :

- Read Only,
- Operate,
- Adjust,
- Debug,
- Program.

How to access the PL7 access management tool

The super user alone may have the necessary privileges to manage user rights.

Step	Action
1	Select the **Access security management** program using the **Start** → **Programs** → **Modicon Télémécanique** command.
2	Enter the access name for the super user : **Supervisor**. By default, access does not require a password.
3	Confirm with **Ok**.

OS-LOADER

Introduction

This tool updates the operating system (OS) of the TSX Micro and TSX Premium PLCs, by downloading via the terminal port.
It is used to :

- display the OS version of the PLC ,
- download the operating system into the PLC system memory.

WARNING

The download operation involves a sensitive stage during which any break in the PLC power supply could make it unusable.

Failure to observe this precaution can result in severe injury or equipment damage.

How to access the OS-LOADER tool

The table below shows the procedure for accessing the OS-LOADER tool.

Step	Action
1	Select the **OS-LOADER PL7…** program. using the **Start** → **Programs** → **Modicon Téléméchanique** command