

OPTIMA PLC

User manual

19/03/2012

OptimaLog
www.optimalog.com



Table of contents

Partie I Introduction	1
1 To what does Optima PLC serve?.....	1
2 General principles	1
Phases of the development.....	1
User's licenses, identification of the customer.....	2
Contained of a project Optima PLC directory.....	2
Contained of a project Optima PLC library.....	3
Compilation and generation of code.....	3
Online modification of the code.....	4
Management of the rights.....	4
3 Installation of Optima PLC.....	4
4 First use: example of creation of a project.....	8
Partie II Programming workshop	10
1 Description of the menus.....	11
Menu =	11
Language software.....	11
Language programming	11
Option License.....	12
Menu Project	13
Project selection.....	13
List of the POUs.....	14
Editor Optima PLC.....	14
File manager.....	15
Window system.....	15
Quit	15
Menu Files	15
New library.....	15
Open a library.....	15
Save all libraries.....	15
Menu Edition	15
Find	15
Replace	16
Next Occurence.....	16
Menu Supervisory	16
Menu Compilation	16
Destination.....	17
Conditions of working for a distant destination.....	17
Parameters for a distant destination.....	18
Full compilation.....	18
Compilation and online modification.....	18
Execution	20
Windows menu	20
Next document	20
Previous document.....	20
Video mode.....	20
Calculator	20
Calendar / Clock.....	21
Format Dates / Hour.....	21
List of the windows.....	21
Editor Options.....	21
General options.....	21
Pascal editor.....	22
ST editor	22
Menu Options	23

General compilation.....	23
System constants.....	24
Settings editor.....	24
Hardware interfaces.....	24
Text output files.....	25
Samples tables.....	26
Miscellaneous.....	28
Startup settings.....	28
Users management.....	28
Windows session for execution.....	28
Windows sessions settings.....	29
Default Files.....	29
Menu Help.....	30
Help.....	30
About Optima PLC Workshop.....	30
2 Description of the icon bar.....	31
Tool bar of project management.....	31
Tool bar of windows display management.....	31
3 Use of the libraries.....	31
General principles of the libraries.....	31
Content of a library.....	32
Task.....	32
Global variables.....	33
Types.....	33
Block of program.....	34
Function block.....	34
Function.....	34
Subgroup.....	34
Trees of the POUs.....	34
"Project" conception tree.....	34
"Libraries" conception tree.....	35
Structure (execution tree).....	35
"Project" Libraries.....	36
Definition order of the POUs.....	36
Properties of the POUs.....	36
Page Declarations.....	36
Page Instructions.....	38
Page type of function.....	40
Page Language.....	40
Page Task.....	40
Page Infos.....	42
Page Properties.....	43
Page Rights.....	45
Table of the visible pages according to the POUs.....	45
Libraries Optima PLC.....	46
Creation of a library.....	46
Menu of libraries management.....	46
Create a POU.....	47
Menu of POUs management.....	47
Creation of Ladder diagrams.....	48
Toolbar of the ladder diagrams.....	49
Management of contacts.....	49
Management of the coils.....	50
Negation.....	51
Management of the diagrams.....	51
Various operations.....	51
Zoom and printing.....	51
Use of the selection.....	51
Use of blocks in a ladder diagram.....	52
Contextual menus of the ladder diagrams.....	53
Example of creation of a ladder diagram.....	54
Creation of FBD diagrams.....	54
Toolbar of a FBD diagram.....	55

Contextual menus of a FBD diagram.....	55
Definition of the type of block.....	56
Use of variables at input and output of block.....	56
Chaining blocks.....	57
Example of FBD diagrams.....	57
Creation of SFC (Grafcet) diagrams.....	58
Toolbar of a SFC diagram.....	59
Contextual menus of the SFC diagrams.....	59
Creation of an action.....	60
Qualifier of action.....	61
Qualifier of action N.....	61
Qualifiers of action S and R.....	62
Qualifier of action L.....	62
Qualifier of action D.....	63
Qualifier of action P.....	63
Qualifiers of action P1 and P0.....	64
Qualifier of action SD.....	64
Qualifier of action DS.....	65
Qualifier of action SL.....	66
Parameters of a boolean action.....	67
Parameters of an action "in situ".....	68
Parameters of a "external action".....	69
Inhibition of an action call.....	70
Create a step commentary.....	71
Manipulation of the actions.....	72
Copy of the elements of a SFC diagram.....	72
Deletion of elements of a SFC diagram.....	73
Example of creation of a SFC diagram.....	73
Reset of a SFC diagram.....	78
Creation of ST block, IL and textual SFC.....	78
Manipulation of the POU's of a library.....	79
Printing	79
printing of diagram.....	79
Printing of POU.....	79
Printing of a library.....	79
Printing of a project.....	79

Partie III Supervisory workshop 79

1 Main menu of the application.....	80
Menu file	80
Menu file - New.....	80
Menu file - Select project.....	80
Menu file - Save.....	80
Menu file - Open.....	80
Menu file - Record as.....	80
Menu file - Quit.....	80
Menu Edit.....	80
Menu edition - Undo.....	80
Menu edition - Cut.....	80
Menu edition - Copy.....	80
Menu edition - Paste.....	80
Menu edition - Delete.....	80
Menu edition - Select All.....	80
Menu Tools.....	80
Menu Create Shortcuts.....	80
Menu User translations - Open a language file.....	81
Menu User translations - New language file.....	81
Menu User translations - AutoExtract page texts.....	81
Menu Apigraf IP Explorer.....	83
Menu Alarms Manager.....	83
Menu Display.....	83
Menu Display - ToolBar.....	83
Menu Display - Function panel.....	83

Menu Display - Object's explorer.....	83
Menu Display - Alignment panel.....	83
Menu Object.....	84
Menu Object - to Add.....	84
Menu Object - to Suppress.....	84
Menu Object - to Import.....	85
Menu Macros.....	86
Menu Macros - local Macros.....	86
Menu Macros - Macros PLC.....	86
Menu Macros - Macros User.....	86
Menu Diagnoses.....	87
TCP / IP PLC data client.....	87
OPC clients.....	87
TCP / IP Alarms data clients.....	88
Measure samples tab linked in memory.....	88
Menu Options.....	89
Menu Options - general Options.....	89
Menu Options - Edition configuration.....	90
Menu Options - Management of the users.....	91
Menu Users.....	91
To change user.....	91
To change the password.....	92
List of the users.....	92
Menu Version.....	92
Menu Help.....	92
2 Creating supervisory forms.....	92
General principles.....	92
Creation of a supervisory form.....	94
Supervisory form general settings.....	94
Style.....	95
Font.....	96
Background.....	96
Grid.....	97
Position / Size.....	98
Options.....	99
Macros.....	101
Supervisory workshop tool bars.....	101
Supervisory objects management.....	102
Create a new supervisory object.....	102
Select a single supervisory object or a group of supervisory objects.....	103
Moving and resizing a supervisory object.....	103
Resizing objects with the zoom function.....	104
Delete supervisory objects.....	104
Bring to front or Send to back an object.....	105
Duplicate objects with the mouse.....	105
Define macros in objects events.....	105
Usual objects for supervisory form designing.....	106
STANDARD tab objects.....	106
Text.....	106
Check box.....	108
Radio group.....	110
Up/Down arrows.....	112
BUTTONS tab objects.....	114
Simple button.....	114
API button.....	117
Push button type #1.....	122
DEL button.....	124
Function button.....	127
Translucent Function button.....	131
Bouton poussoir industriel.....	135
Push button.....	138
INPUT BOX tab objects.....	139
Standard input box.....	139

Custom input box.....	144
Date / Time input box.....	149
Combo box.....	151
CONTROLS tab objects.....	155
Page control.....	155
Scroll box.....	157
HTML Viewer.....	159
Panel	160
List Box	163
Click zone.....	165
DRAW tab objects.....	167
Shape	167
Line	169
Gradient.....	170
GRAPHIC tab objects.....	172
Graph	172
Histogram.....	183
Scale	186
Pie graph.....	187
ANIMATIONS / PICTURES tab objects.....	190
Animation.....	190
Pictures animated.....	193
Text parading.....	196
Animated bitmap.....	197
Picture	199
GAUGES tab objects.....	201
Gauge	201
Vu meter.....	203
Progress bar.....	205
Simple circular gauge.....	206
Advanced Circular gauge.....	209
Analog Vu meter.....	213
CURSORS / TRACKBARS tab objects.....	216
Trackbar.....	216
Advanced TrackBar.....	218
Scroll bar.....	221
SWITCHES tab objects.....	223
ON/OFF Switch.....	223
Lever switch #2.....	225
Emergency stop button.....	225
SELECTORS tab objects.....	226
2/3 positions Selector.....	226
2/3 positions Selector Without Label.....	230
INDICATORS tab objects.....	230
Indicator type 1.....	230
Simple indicator.....	231
Double indicator.....	233
3 colors column indicator.....	235
DEL indicator.....	237
EXCEL tab objects.....	239
EXCEL link.....	239
CUVE / FLOODGATE tab objects.....	242
Cuve	242
Floodgate.....	245
Floodgate bitmap.....	247
HOUR / DATE tab objects.....	249
Display Time.....	249
Display Date.....	250
Calendar.....	252
Date / Hour input box.....	254
ALARMS tab objects.....	254
Alarm history.....	254
RECORDING tab objects.....	255
Recording of data.....	255

Curve Tape recorder.....	258
MULTIMEDIA Tab objects.....	272
Média player.....	272
Flash ShockWave player.....	274
OTHERS tab objects.....	276
Recipe	276
Billposters 7 segments.....	279
OLE Container.....	281
Event Manager.....	283
Objects explorer.....	285
Use of the explorer of objects.....	285
To conceal an object in conception.....	285
To conceal all objects in conception.....	285
To annul the masking of an object in conception.....	286
To annul the masking of all objects in conception.....	286
To configure an object.....	286
To suppress an object.....	286
Open a supervisory form from another.....	286
Open a supervisory form with parameters.....	287
Closing a supervisory form with a button.....	287
Passwords management.....	287
3 Objects personalized.....	288
Registration of objects Personalized.....	288
Deletion of objects personalized.....	288
Import of objects personalized.....	289
4 Explorer of objects.....	290
Explorer of objects.....	290
5 Use of the macros.....	291
Local macros.....	291
BringToFrontSuperGWindow (local Macro).....	291
ChnMdp (local Macro).....	291
ChnUserLanguage (local Macro).....	291
Close (local Macro).....	292
CopyPageBitmap (local Macro).....	292
DiagSocketPLC.....	292
ExecProg (local Macro).....	293
HideCursor (local Macro).....	293
HideSuperGWindow (local Macro).....	293
PrintPage (local Macro).....	294
RunHelp (local Macro).....	294
RunHelpIndex (local Macro).....	294
RunHelpTopic (local Macro).....	294
ShowCursor (local Macro).....	294
ShowSuperGWindow (local Macro).....	295
SuperG (local Macro).....	295
VisuText (local Macro).....	295
WFCopy (local Macro).....	295
WFCreateDirectory (local Macro).....	296
WFDelete (local Macro).....	296
WFDeleteDirectory (local Macro).....	296
WFRename (local Macro).....	296
WFSetFileAttr (local Macro).....	296
WNetSChangeFileName (local Macro).....	296
WNetSCopyFileRemote (local Macro).....	297
WNetSCreateDirectory (local Macro).....	297
WNetSDeleteDirectory (local Macro).....	297
WNetSReadAndMoveFile (local Macro).....	297
WNetSWriteAndMoveFile (local Macro).....	297
WNetSReadFile (local Macro).....	297
WNetSWriteFile (local Macro).....	298
WNetSSetFileAttr (local Macro).....	298
External or distant macros (executed on the PLC).....	298
BEEP (Macro PLC).....	298

DECR ###(Macro PLC).....	298
INCR ###(Macro PLC).....	298
INVSA ###(Macro PLC).....	298
SET...= (Macro PLC).....	299
User macros.....	299
Generalities.....	299
Development of a User DLL.....	299
User DLL example.....	303
Using a user DLL in a project.....	305
Unload a user DLL in a project.....	305
User DLL list.....	305
6 Translations & Languages management.....	306
Language files.....	306
Creating a new language file.....	306
Using language files in the supervisory forms.....	306
Selecting a language.....	306
Translation example.....	306
7 Users Management.....	306
Generalities.....	306
8 Execution of the supervisory forms.....	307
Files create in a project of supervisory.....	307
Execution of one page since the workshop.....	307
Execution of one page with its icon.....	307
9 Virtual keyboard.....	307
Use of the virtual keyboard.....	307
10 Use of DDE ties	308
General principles.....	308
Example: Creation of a DDE tie with MICROSOFT® Excel.....	308
Authorization of modification of variables PLCs via a DDE tie.....	308
11 Access to the variable PLCs since a program writes in DELPHI.....	308
General principles.....	308
Creation of a DELPHI5 program using the USERAPIPLC packet.....	309
Accessible functions since the USERAPIPLC packet.....	309
LinkToAPI function.....	309
IsLinkedToAPI function.....	309
AllLinksOK function.....	310
UnLinkToAPI function.....	310
ExecMacro procedure.....	310
PLCState function.....	311
12 Variables local OPTIMA VIEW.....	312
General principles.....	312
Description of the local variables.....	312
Definition of the number of local variables.....	312
Visualization of the local variables.....	312
13 OPC server: ServApiOPC.....	312
Generalities.....	312
ServApiOPC application.....	313
Access to the data of the OPC server.....	314
Name of the OPC server.....	314
Definition of a group.....	314
Definition of an item.....	314
Definition of the data in read/write.....	314
14 Use of the workshop in independent mode.....	315
Installation and starting of the application of Supervisory.....	315
Installation of the application.....	315
Starting of the application.....	316
Window of project selection.....	317
Selection of a project.....	317

1 Generalities	318
Starting of the application.....	318
Stop of the application.....	319
Users management.....	319
Status of the application.....	320
Performances and real-time.....	320
2 Description of the environment.....	322
Main window.....	322
Recall of the window.....	323
Statusbar.....	324
3 Description of the menus.....	324
Menu Control.....	324
Menu Options.....	324
Option Licenses.....	327
Option to Leave.....	327
Menu Users.....	327
Menu PLC.....	327
Option Supervisory.....	328
Option Tasks.....	328
Graficets option.....	328
Option standard Variables.....	329
Option Expressions.....	330
Option Explorer.....	331
Option Outputs Text.....	331
Option Serial Ports.....	332
Drivers option.....	333
Menu Windows.....	333
Menu Diagnoses.....	334
Debugger option.....	335
Option Performances.....	335
Option Log.....	337
Option OPC server.....	338
Option Lists of sockets.....	338
Option to "Generate a registration of diagnosis".....	339
Menu Help.....	340
4 Possible problems to the starting of the runtime.....	341
No "retain" data file at the starting of the application.....	341
Modification of the structure of the "retain" data.....	341
Absence of data concerning the users.....	342
Problem of starting of a hardware driver.....	343

Partie V Management of the users 343

1 Presentation	343
2 Menu Users	344
Option Change user.....	344
Option to Change the password.....	344
Option Lists some users.....	345

Partie VI Programming 345

1 Data types	346
Integer.....	346
Real.....	346
Duration.....	347
Time stamp.....	347
Characters string.....	347
Character.....	348
Bits string.....	348
Generic types.....	348
Derivative types.....	349
Data Structure.....	349

Enumerated Types.....	349
Ranges of values.....	350
Arrays	350
2 Preset variables in Optima PLC.....	350
Tables of variables.....	350
Particular cases.....	351
A Alarms.....	351
DT temporizations.....	351
Various variables.....	352
3 Declarations	352
Declarations of variables.....	352
Attributes of variables.....	353
POU instances.....	353
4 Constant values.....	354
Numeric constants.....	354
Constants characters strings.....	354
Constants of duration.....	355
Constants of date.....	355
Typed constant values.....	355
5 Initialization of variables.....	356
Initialization of types and simple variables.....	356
Initialization of types and multi-elements variable.....	356
6 Programming in structured text language (ST).....	357
Expressions.....	357
Instructions.....	358
Affection	359
Call of function.....	359
7 Programming in instructions list (IL).....	359
Operators, modifiers, operands.....	359
Call of functions.....	361
Call of function blocks.....	361
8 Programming in Apigraf language.....	362
Typical boolean instruction.....	362
Condition (boolean expression).....	363
Example of conditions and actions.....	364
Edge detection.....	366
Boolean conditions evaluation.....	368
Numeric operations.....	368
Writing on output text file.....	369
Grafcet instructions.....	369
Declaration of a sequence variable	369
Definition of the code of a sequence.....	369
Definition of a sequence step.....	370
End of a sequence.....	370
Jump to a step of the sequence.....	370
Examples of sequences	370
Transposition of a Grafcet in Apigraf language.....	372
Differences with the original Apigraf language.....	372
9 Advanced programming.....	374
Techniques of time management.....	374
Display of information, of errors, of confirmation demands.....	375
Use of the samples tables standards.....	376
Use of the output text files.....	376
Partie VII "On line" modifications.....	376
1 Application	376
Modifications of the program of the application.....	376
Modes of loading of the modifications.....	377
2 Modifications of the supervisory.....	378

Partie VIII	Examples	378
1	Creation of a program.....	378
2	Example 1: Motor starting.....	379
	Working.....	379
	Declarations.....	379
	ProgrammeST.....	380
	ProgrammeFBD.....	380
	ProgrammeLD.....	381
	ProgrammeSFC.....	381
3	Example 2: Automatic door.....	382
	Working.....	382
	Declarations.....	383
	Working.....	383
	Validation of the code.....	384
4	Other Optima PLC examples.....	385
Partie IX	Drivers Optima PLC	386
1	Advantech	386
	Card Advantech PCL-728.....	386
	Parameters.....	386
	Usage	387
	Card Advantech PCI-1710.....	387
	Parameters.....	388
	Usage	388
	Status values.....	389
	Card Advantech PCI-1711.....	389
	Parameters.....	390
	Usage	390
	Status values.....	391
	Card Advantech PCI-1715.....	391
	Parameters.....	391
	Usage	392
	Status values.....	392
	Card Advantech PCI-1723.....	393
	Parameters.....	393
	Status values.....	393
	Card Advantech PCI-1733.....	394
	Parameters.....	394
	Status values.....	394
	Card Advantech PCI-1734.....	394
	Parameters.....	395
	Status values.....	395
	Card Advantech PCI-1747U.....	395
	Parameters.....	395
	Usage	396
	Status values.....	396
	Card Advantech PCI-1751.....	397
	Parameters.....	397
	Status values.....	397
	Card Advantech PCI-1762.....	398
	Parameters.....	398
	Status values.....	398
	Card Advantech PCI-1780U.....	399
	Parameters.....	399
	Status values.....	399
	Modules Adam.....	400
	Adam 4017, 4018.....	400
	Parameters.....	400
	Status values.....	400
	Adam 4050, 4053, 4055.....	401

Parameters.....	401
Status values.....	402
2 Hilscher	402
Hilscher generic CIF board.....	402
Installation.....	402
Parameters.....	402
Steps of the parameterization.....	403
Status values.....	413
Stations	414
Hilscher - generic Station deportees	414
Parameters.....	414
Securities of statute.....	415
Hilscher - Station deportees 16I 16O	415
Card Hilscher CIF CAN Layer 2.....	415
Card Hilscher CIF 50-in Modbus / TCP IO mode (server).....	415
Installation.....	416
Parameters.....	417
Status values.....	417
Card Hilscher CIF 50-En Modbus/TCP message mode (client).....	417
Installation.....	417
Parameters.....	418
Status values.....	418
3 Heidenhain	419
IK220 card.....	419
Installation.....	419
Parameters.....	419
Usage	420
Status values.....	420
4 ICP DAS	421
Carte ICP DAS PCI-P16R16.....	421
Parameters.....	421
Status values.....	421
5 Phoenix Contact.....	421
Modbus / TCP module marks FL IT 24 BK-PAC.....	421
Parameters.....	422
Usage	422
6 Port I/O PC	423
Port I/O PC - 8 digital inputs.....	423
Parameters.....	423
Usage	424
Port I/O PC - 8 digital outputs.....	424
Parameters.....	424
Usage	424
7 Simulation	425
Standard simulation driver.....	425
Parameter.....	425
Usage in the supervision.....	425
Access by shared storage zone.....	426
Driver of simulation Solid Dynamics Motion.....	426
Parameters.....	426
Usage in the supervision.....	426
Access by shared storage zone.....	427
8 Wago	427
Modbus/TCP module 750-341, 342, 352, 841, 842.....	427
Parameters.....	427
Usage	428
Status values.....	429
9 Network standard Ethernet.....	429
Generic Modbus/TCP module.....	429
Parameters.....	429
Status values.....	431

Partie X Versions history of the software components	431
1 OptimaPLCRuntime.exe (runtime Optima PLC)	431
V1.4.10	431
V1.5	431
V2.0	431
V2.1	431
V2.2	431
V2.2.1	432
V2.5	432
V2.6	432
V2.8	432
V2.8.2	432
V2.9.4	432
V2.9.5.5	432
V2.10	432
V2.10.8	432
V2.10.9	432
V2.11	432
V2.12.7	432
v2.13	432
v2.14	432
2 Drivers.exe (selection of the drivers in Optima PLC)	432
v1.4.9	432
V2.3	432
V2.10.7	432
V2.10.8	433
V2.12.7	433
v2.13	433
v2.14	433
3 OptimaPLCWorkshop.exe (atelier Optima PLC)	433
V1.4.9	433
V1.4.10	433
V1.5	433
V1.5.1	433
V2.0	433
V2.1	433
V2.1.2	433
V2.1.3	433
V2.2	433
V2.2.1	433
V2.3	434
V2.4	434
V2.5	434
V2.6	434
V2.7	434
V2.8	434
V2.8.2	434
V2.9	434
V2.9.1	434
V2.9.4	435
V2.9.5	435
V2.9.5.5	435
V2.10	435
V2.10.1	435
V2.10.6	435
V2.10.7	435
V2.10.9	435
V2.11	435
V2.12	435
V2.12.7	435
v2.13	436

v2.14	436
4 SuperGW.exe Optima View (Supervisory workshop).....	436
V4.53	436
V4.54	436
V5.0	436
V5.1	436
V5.2	436
V5.3	436
V5.4	436
V5.8	436
V5.10	436
V5.11	436
V5.12	437
V5.13	437
V5.14	437
V5.15	437
V5.16	437
V5.17	438
V5.34	438
V5.38	439
5 ServAPIOPC.exe (OPC server Apigraf IP).....	439
V5.31	439
V5.32	439
V5.50	439
6 OptimaAlarm.exe (Optim'Alarm).....	439
V1.1	439
V1.2	439
V1.3	439
V1.5	439
V1.6	439
V1.8	440
V1.9	440
V1.11	440
V1.12	440
V1.20	440
V1.23.1	440
7 Tracks.exe (Optima track).....	440
V1.0	440
V1.1	440
V1.4	440
V1.5	440
V.1.5.1	440
V1.6	440
V1.7	440
V1.8.1	441
V1.9.1	441

Partie XI Frequent questions 441

1 General questions.....	441
Why use the PC as a PLC ?.....	441
Does Optima PLC permit to program a PLC of the market?.....	441
Is Optima PLC a simulation tool ?.....	441
Questions concerning the real-time aspects.....	441
Is Optima PLC a real-time system?.....	441
Can Optima PLC function without the real-time kernel?.....	441
Why is the real-time kernel deactivated by default?.....	441
Does Optima PLC require the supplementary hardware to function in real time?.....	442
I did think that it was possible to make real-time with Windows, I don't see what Optima PLC brings besides?.....	442
Why not use a real-time system of the market rather?.....	442
Does Optima PLC permit to make real-time with all PC?.....	442
Does Optima PLC operate resources of the PC multiprocessors?.....	442

Does Optima PLC permit to make real-time with all interfaced hardwares?.....	442
What performances does one get without the real-time kernel?.....	443
Is it possible to pass from a language of the standard automatically to another?.....	443
Is it possible to modify an application without stopping it?.....	443
I did already work with Apigraf, can I migrate an Apigraf project easily under Optima PLC?.....	443
Where can I download the last version of Optima PLC?.....	443
Does Optima PLC respect the standard IEC 61131-3?.....	444
2 Technical questions.....	444
Hardware.....	444
What configuration is necessary to make Optima PLC operate ?.....	444
What can arrive if I don't have memory enough on the PC?.....	444
How to make.....	444
How to get some information on the user's license of Optima PLC?.....	444
How to limit the hazard that the user corrupts data or the system under Windows?.....	444
How to prevent the user to change operating parameters in the runtime?.....	444
How to use timers in Optima PLC?.....	444
How to display some warning messages for the user?.....	444
How to force a SFC diagram to its initial step ?.....	444
How to write data in a file to reread them with a spreadsheet?.....	445
How to enter values of parameters on my touch panel?.....	445
How to make the start of the PC start my application automatically?.....	445
How to make my Optima PLC application communicate with another application?.....	445
How to make display fault messages simply ?.....	446
Resolution of problems.....	446
I define some mnemonics on variables standards but they don't appear.....	446
I have a compiling error on a function name described in the documentation.....	446
A blue screen is displayed or the PC restarts during the execution of the runtime.....	446
I have a problem with Optima PLC and I don't find the solution in the documentation, who may I contact ?.....	447
Various.....	447
How can I reduce the compilation time of my application?.....	447

Partie XII Appendices

447

1 Optima APIFTP Servers.....	448
Main page.....	448
definition of the sharings.....	448
2 Codes of APIFTP result.....	449
opInProgress.....	449
opOK.....	449
opProtocolError.....	449
opConnectionError.....	449
opConnectionAborted.....	450
opConnectionTimeOut.....	450
opNoRessource.....	450
opDestinationUnreachable.....	450
opVersionAnswerTimeOut.....	450
opSourceNotFound.....	450
opUnableToOpenSource.....	450
opUnableToReadSource.....	450
opExistFile.....	450
opPathNotFound.....	450
opAccessDenied.....	450
opUnableToCreateDest.....	450
opUnableToWriteDest.....	450
opFileSizeError.....	451
opUnableToRenameDest.....	451
opSourceIsDir.....	451
opSourceIsVolumID.....	451
opSourceIsWriteProtected.....	451
opUnableToDeleteSource.....	451
opUnableToRenameSource.....	451
opUnableToCreateDir.....	451

opUnableToDeleteDir.....	451
opDirAlreadyExists.....	451
opUnableToGetVersion.....	451
opUnknownParameter.....	451
opUnableToExecFile.....	451
opProtocolVersionError.....	451
opError	452

Index	453
--------------	------------

1 Introduction

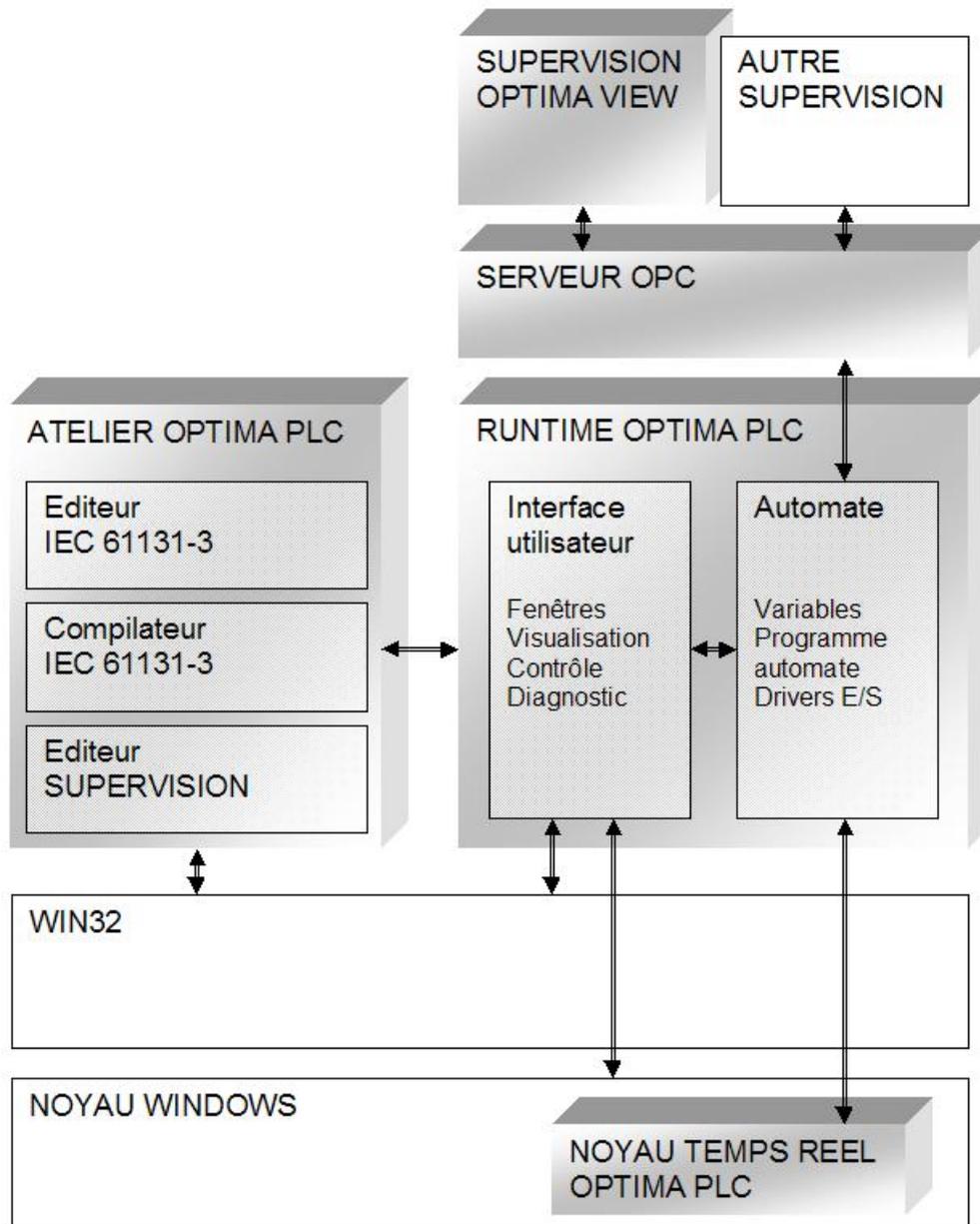
1.1 To what does Optima PLC serve?

The main destination of Optima PLC is to assume the role of industrial Programmable Logical Controller (PLC) on a standard PC. It also permits to build quickly a graphic supervisory associated to this PLC, as well as of other supervisory tools proposed by Optimalog.

Optima PLC includes a development workshop in conformity with the standard of programming of PLCs IEC 61131-3, a supervisory workshop, and a system permitting to finalize and to execute the application under Windows in real-time.

1.2 General principles

The diagram following shows the general structure of Optima PLC components.



1.2.1 Phases of the development

The use of Optima PLC is made schematically in three phases :

- The programming of the application

The dedicated development workshop permits to program and to organize the "units", named POU's, according to the principles of the norm IEC 61131-3. These POU's are regrouped within a "project." The workshop also permits to construct a supervisory (human/machine interface) associated to the program.

- The compilation of the application

When the application is developed, the programmer uses the compilation function. This operation allows the workshop to generate the optimized (there is not interpretation of the instructions) executable code.

If compiling errors are detected during this phase, they are pointed to allow the programmer to correct them.

- The execution of the application

Finally, the application is executed with the dedicated real-time kernel of Optima PLC and the runtime, allowing it to optimize the launching of the tasks defined by the programmer.

1.2.2 User's licenses, identification of the customer

The right of use of the software is assured by the presence of a key USB inserted in the PC during the working of the workshop and the runtime.

The two types of possible key are below :



Ithea key



Optimalog key

The key can contain a license for the workshop, or a license for the runtime, or a license for both. The key can also contain a license for the other Optimalog software.

A user's license includes a client code (numeric), unique for every customer of Optimalog. This code is reported in the objects created with Optima PLC, what permits "to sign" the creation of it, and to manage the rights that the author of a program wishes to grant or no to the other customers: visualization, modification, etc.

The key Optimalog can contain some files, in particular the installation program.

In the protective key absence, Optima PLC can be used with some limitations :

- Your identifying is "Unknown."
- The workshop is usable during a maximum of 30 days.
- The time of execution of the runtime is limited to one hour, after what it must be launched again.
- A warning message is displayed at workshop and runtime launching.

Attention: when the trial period of 30 days is finished, Optima PLC cannot be executed anymore without protective key.

1.2.3 Contained of a project Optima PLC directory

A project is a directory (with the name of the project) in which all files necessary to the creation and to the execution of the PLC program are stored.

The directory of the project at least contains the following files :

Main.opl	Library containing the main programs (by default)
xxx.opl	Libraries defined or imported by the user in the project
NomDuProjet.ini	Settings file
NomDuProjet.inb	Settings file backup
OptimaPLC.dsk	Development workshop desk file

After compilation and execution :

OptimaPLC.prg	executable program
---------------	--------------------

Debug.dat	output text file, automatically opened to the starting of the runtime and saved and closed at the time of the stop of the runtime.
Debug xxxx.xml	File containing the complete data of the PLC in xml format. The zone xxxx corresponds at the date and hour of creation of the file. This file can be created to the user's demand (menu Diagnoses of the runtime) or can automatically be generated at the exit from the application if this one has been compiled with the option "debug information".

Supervisory: the files are placed in the PSGW subdirectory of the project

ProjectName.psgw	Main supervisory form
other files psgw	Supervisory forms called from the main form

Files of desk description, in the DESK subdirectory of the project :

Admin.dsk	description of the administrator's office
xxx.dsk	description of the office of the user xxx, if the management of the users is activated.

Files of the variable "retain" of the application, in the RETAIN subdirectory of the project :

.retain files	Files saved every 5 minutes (by default). The name corresponds to the date and time of the back-up.
---------------	--

1.2.4 Contained of a project Optima PLC library

A library of project is stored as a .OPL file (Optima PLC Library). A project Optima PLC can contain several libraries. Some standard libraries are often used and charged systematically (system.opl, standard.opl, etc).

A library is constituted of POU's of different types :

	Task	POU of execution schedule of the program blocks
	Global variables	POU of declarations of available variables for the set of the programming POU's
	definitions of types	POU of declarations of type of available variables for the set of the programming POU's
	block of program	Main programming POU
	function block	Sub-program, with or without return value, that memorizes the state of its internal variables between 2 calls, so for a same set of input parameters the returned result by this block can be different (example of the PID: for a same order the output of this regulation block evolves until it reaches the order)
	function	Subroutine whose return value remains identical for a same set of input parameters (no memorization of the internal variables)
	subgroup	Library structuring POU.

A project contains one or several libraries, each containing one or several tasks, calling one or several programs that can themselves call function blocks and functions.

1.2.5 Compilation and generation of code

To be able to execute the developed application, Optima PLC proceeds in two exploratory stages :

- Verification of the code and control of the consistency of the POU's: to this stage, the application is controlled entirely by Optima PLC.

- Generation of the executable code: the POU's are compiled directly in executable optimized code. The result of this compilation is the OptimaPLC.prg file placed in the directory of the application.

1.2.6 Online modification of the code

Optima PLC permits to do some modifications of the application and to put them in service "on line", i.e. without interrupting the working of the application.

For it, Optima PLC proceeds as follows :

- Verification of the code and control of the consistency of the POU's (as for the normal code generation).
- Verification of the modifications done since the starting up of the application. To this stage, some modifications detected can interrupt the new code on line activation.
- Generation of the executable code : the generated code is executable directly by the CPU and optimized.
- Loading and setting in service of the modified executable code, with minimal interruption of the application (less than one millisecond).

1.2.7 Management of the rights

The author of one POU "sign" his creation with the code contained in the protective key. When another customer (having a different key, with another client code) wishes to visualize or to modify this POU, his rights can be limited to the only actions allowed by the author.

In this manner, an integrator can for example, if it wishes it, protect his intellectual property against the consultation, or prevent all modification to his code during the period of guarantee of his installation.

See ["Rights Page"](#) in the POU options.

1.3 Installation of Optima PLC

Optima PLC must be installed on a PC whose operating system is Windows XP or Vista.

If you have an Ithea key and if it is a first installation, the driver of the USB Ithea protective key will be installed simultaneously. **Important : the key Ithea must not be inserted before the end of the first installation.**



Ithea key

You can do the installation from different supports :

<p style="text-align: center;">Optimalog key</p> 	<p>The key acts as protective key and support of the installation files. Insert the key in an USB connector of the PC. Windows automatically conducts the installation of the key, but doesn't run the installation software of Optima PLC. Open the corresponding volume to the key with the files explorer, then run the installation program.</p>
<p style="text-align: center;">CD ROM</p>	<p>Insert the CD ROM in the reader. The automatic starting proposes you the installation of Optima PLC as well as other useful third-party software.</p>
<p style="text-align: center;">Downloaded installation program</p>	<p>Execute the program and follow the described indications below.</p>

If you don't possess a protective key, Optima PLC will function in trial mode.

Importance of the Windows session

Attention, it is counseled to do the installation from a Windows session in which you have administrator's rights. The whole installation will be done without problem, and the directories of examples and projects will be placed in :
"Documents shared \Optima PLC."

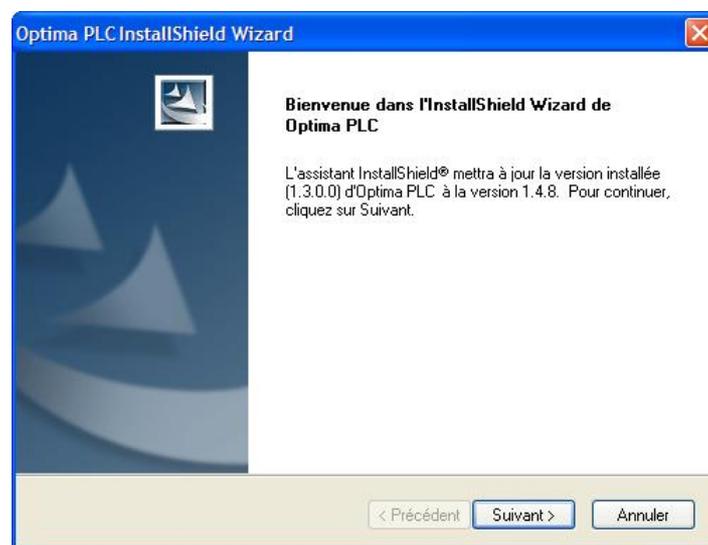
Besides, the installation will be then accessible to all users.

On the other hand, if the installation is done from a session not having administrator's rights :

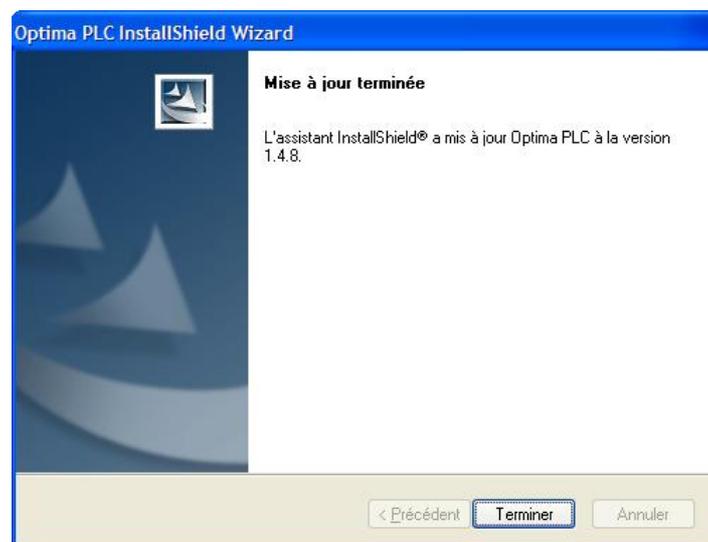
- some files won't be able to be installed in the directories of Windows (OPC server for example).
- the installation in progress will only be accessible to the current user
- the directories of examples and project will be placed in :
"My documents\Optima PLC."

Starting of the installation

Run the installation program.



Once the finished installation, a reboot is proposed, or a screen as the next one is displayed if it is not the first installation :

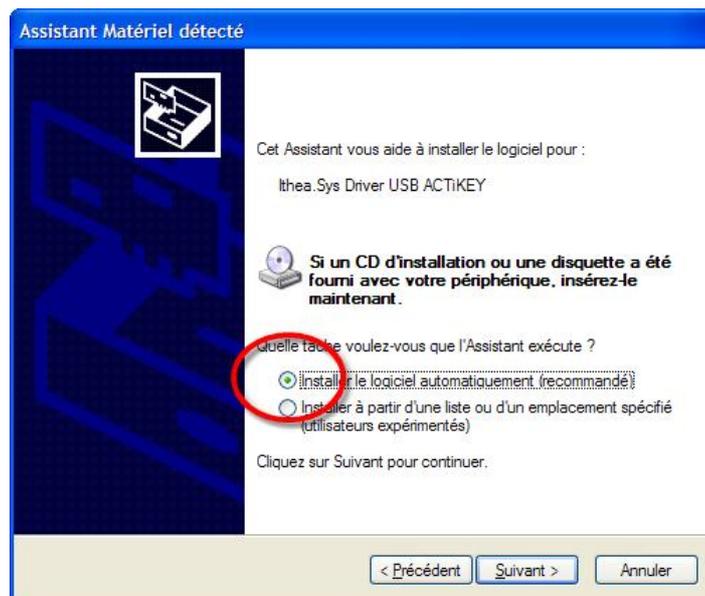
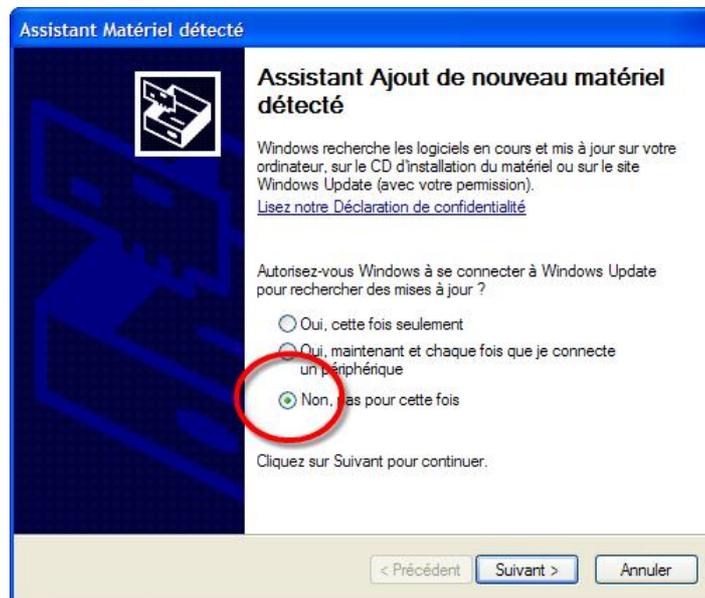


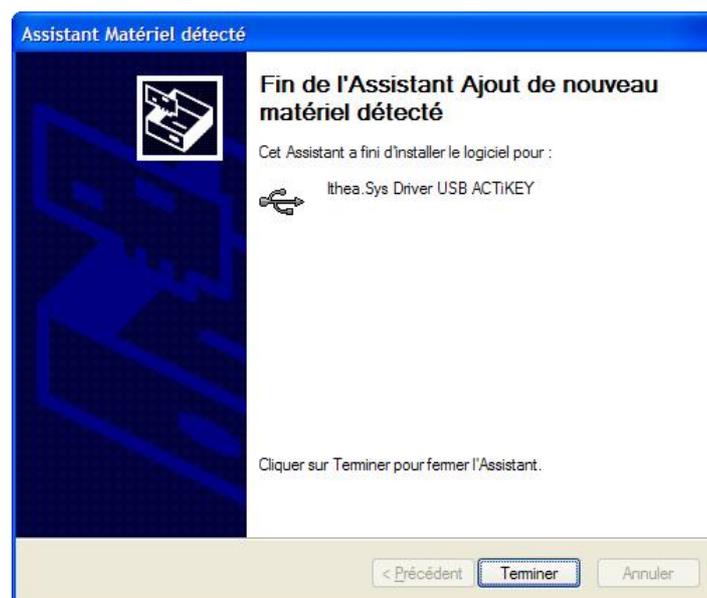
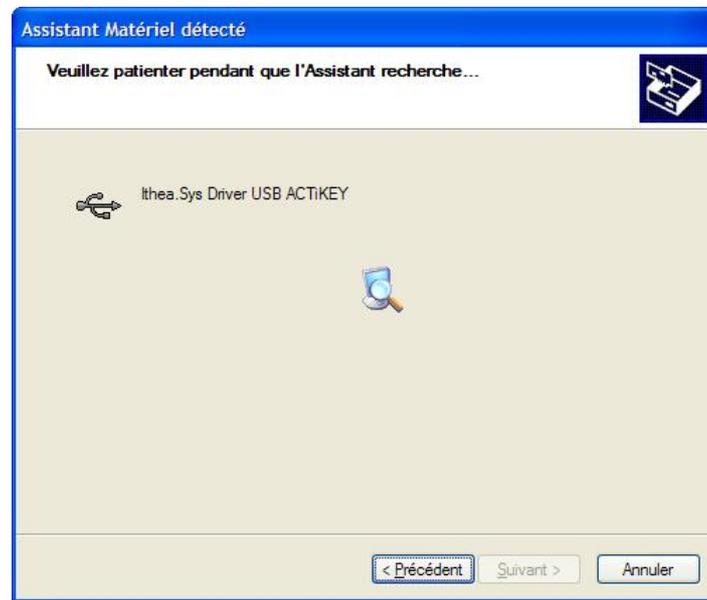
At the time of the first installation, a reboot is necessary in order to make operational the OPC server integrated to Optima PLC. If the installation done is not the first, it is possible to run the software immediately from the Start menu.

Installation of the Ithea key

If you don't have an Ithea key, the installation is now finished. If you possess this type of key, and if it is a first installation, Windows must finalize the installation of the Ithea USB protective key : after the reboot of the PC, insert this key now in one of the USB connectors.

The following windows are displayed, press "next" buttons every time :





The installation is now finished, the software can be executed from the Start menu.

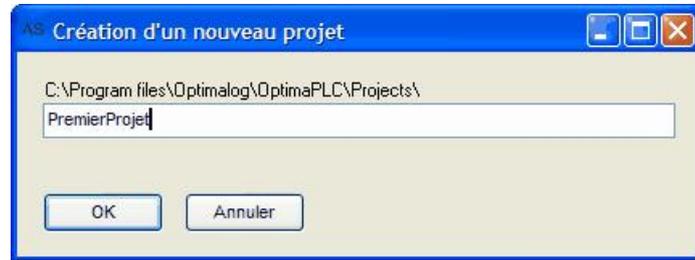
1.4 First use: example of creation of a project

Start the development workshop of Optima PLC (double-click on the icon present on the desktop).

At the starting, a dialog box permits the selection of project.

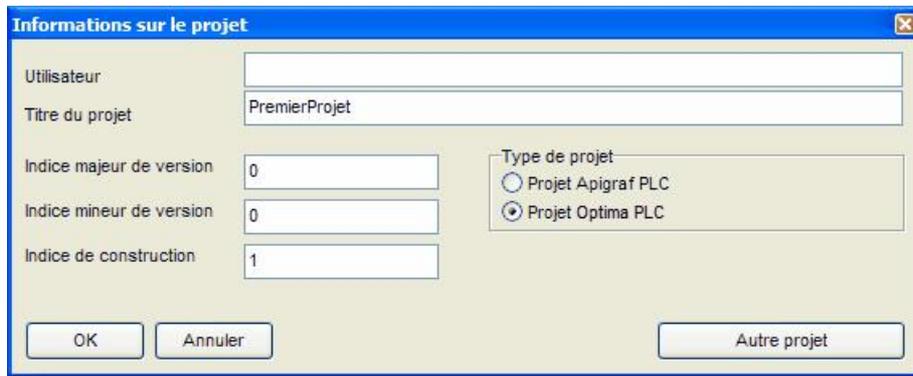
In the tree that appears, select the directory in which the project must be created (normally "Projects") then click on the button "New project".

The window that appears then permits to give a name to your project :



Above, in the example the created project is "PremierProjet".

After having confirmed the creation of the project, a new window permits to enter the information relative to this project :

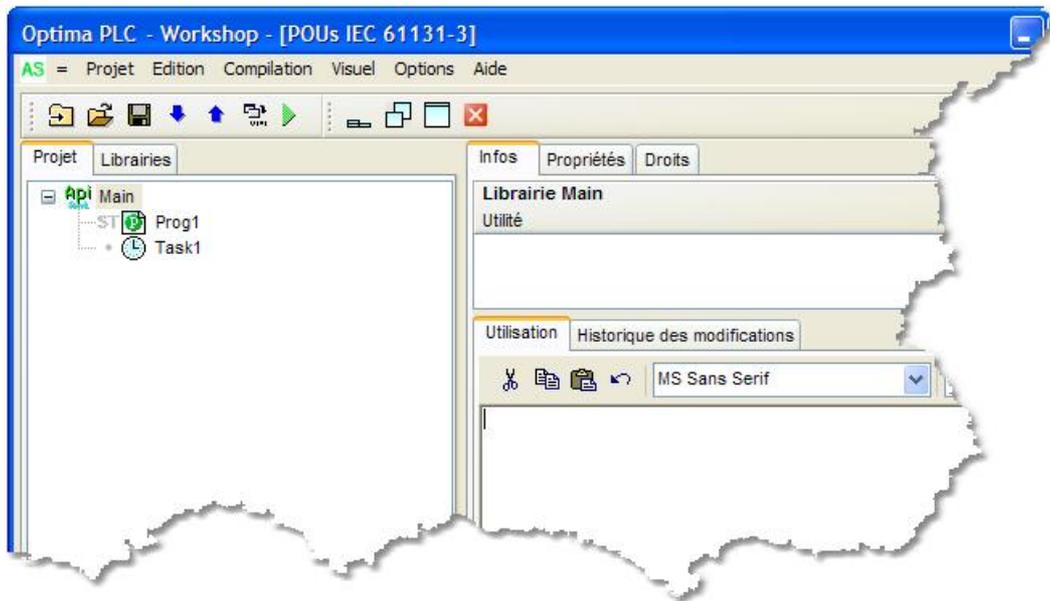


- The user's name
- The title of the project
- The type of project (choice between a project Apigraf PLC and a project Optima PLC)
- The version number (composed of 3 indications, V0.0.1 at the creation)

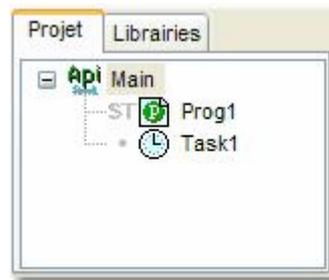
The project is now created, the programming is the following stage.

The environment of development is composed :

- Of the publisher's menu
- Of a bar of icons for the most current functions (example: Save all, Execute.)
- Of tabs permitting to reach the organization of the program and the libraries of standard programming object of Optima PLC
- Of tabs to program the different POU's of the project (the number of tabs varies according to the block selected in the panel of the libraries)



At the time of the creation of the project, some elements are automatically inserted in the tab "Project " :

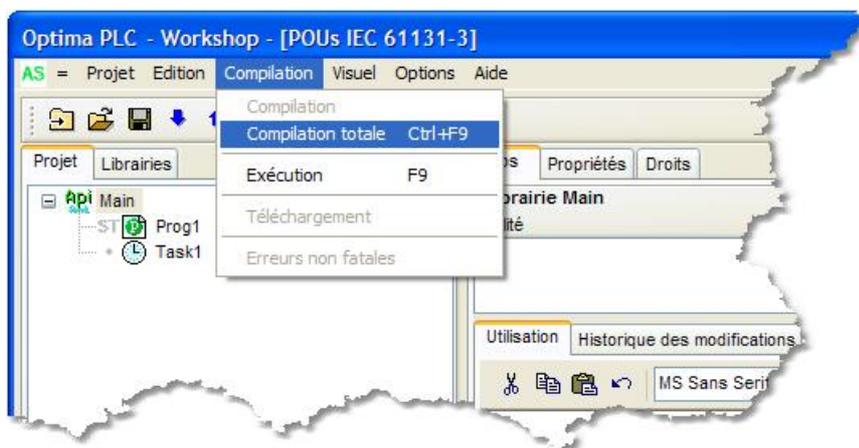


A "main" folder corresponds to default program library.

In this library, two POUs are created. They compose the minimum program (that is that the program can already be compiled and can be executed) :

- A task: this POU permits to define the blocks of programs that must be executed, their priority as well as the period or the event that give rhythm the execution.
- A block of program: to write the program (by default, programming in ST language)

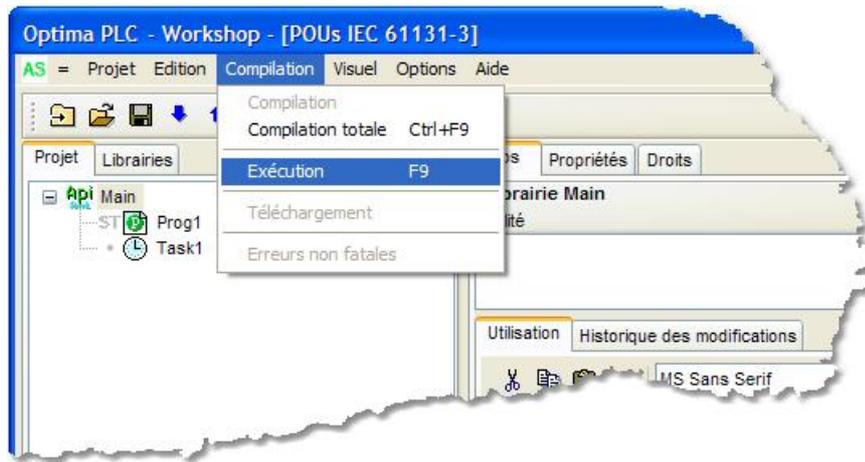
From now the program can be compiled executed (even though at present this program doesn't make anything). To compile the program, select the menu "Compilation → Full compilation" or click on the suitable icon.



OR 

A message is displayed: "Please wait, compilation in progress". When the compilation is finished, a new message indicates if this one has been achieved with or without error. At present no line of program having been written in the project, the compilation ends without error.

To execute the program, select the menu "Compilation → Run" or to click on the suitable icon.

or 

The application is then started, the environment of the Runtime appears :



It is composed :

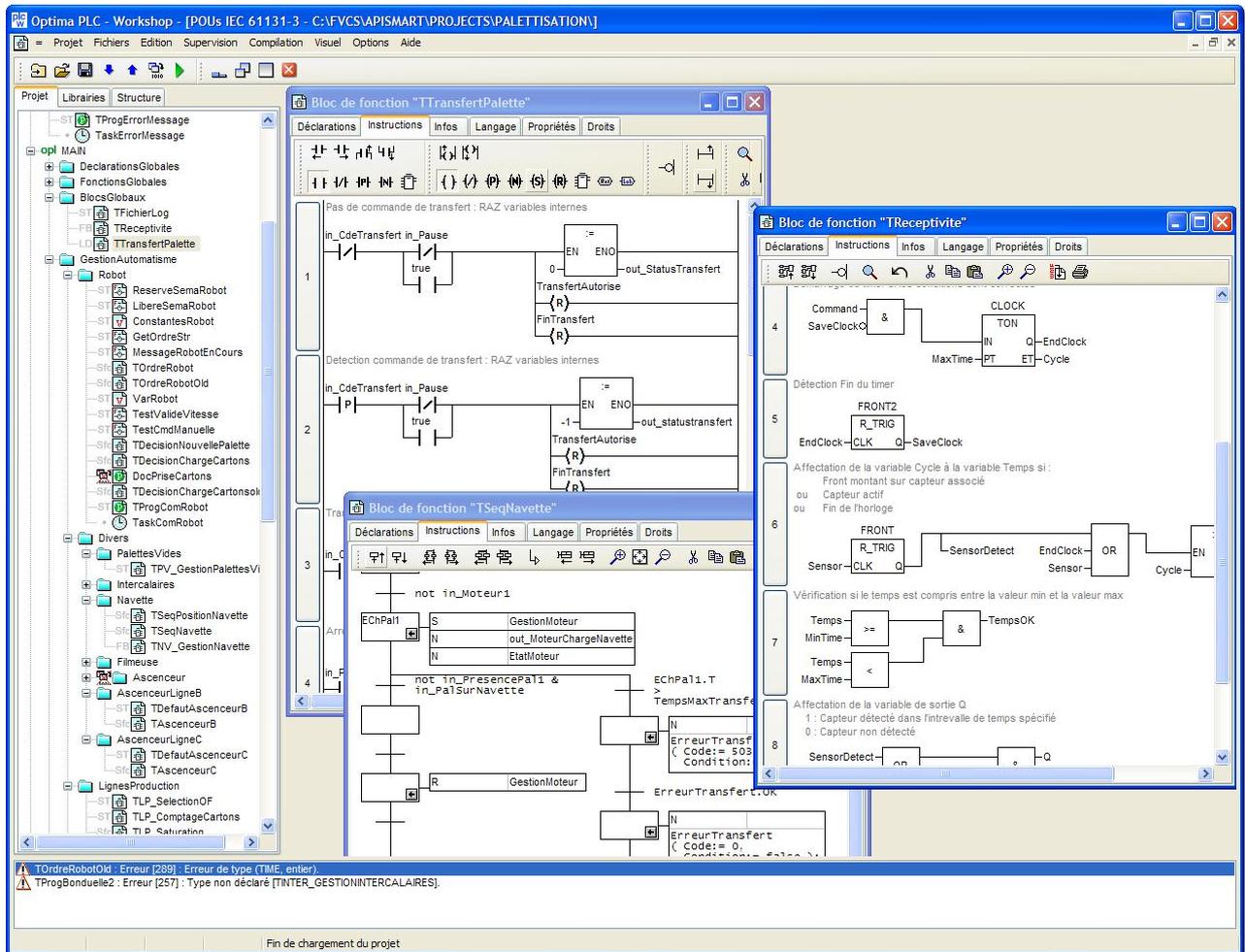
- Of a menu
- Of a status bar (below)
- The center of the window is dedicated to the runtime windows (example : view of the variables)

To leave the runtime, select the menu "Controls → Quit" and confirm the demand of application stop

The environment of Optima PLC is now more familiar to you. The following chapters provide more details on the programming and on the use of the workshop and the runtime.

2 Programming workshop

This chapter retails the use of the programming workshop Optima PLC, that permits to define the POU's of an application according to the recommendations of the norm IEC 61131-3.



2.1 Description of the menus

2.1.1 Menu =

2.1.1.1 Language software

This option permits to select the language used for the displays in the workshop.

2.1.1.2 Language programming

The change of programming language is currently only available when the project is a project of Apigraf type.

2.1.1.3 Option License

This option permits to display the information concerning the user's license of the software.

Optima PLC - Licences d'utilisation

Optima PLC

Localisation

Aucune clé

Clé USB locale

Clé USB distante

PC

No de série 105277

Type de licence Gold

Numéro de licence 612002

Début/fin de validité 09/12/2006

Nom utilisateur Dupont Automatism

Code utilisateur 113

Fin de maintenance logicielle

Fin d'assistance technique

Atelier logiciel

Runtime

Mono-application

OPTIMA View

Nb Max I/O TOR

Nb Max I/O Ana

Actualiser

Fermer

Modification Licences

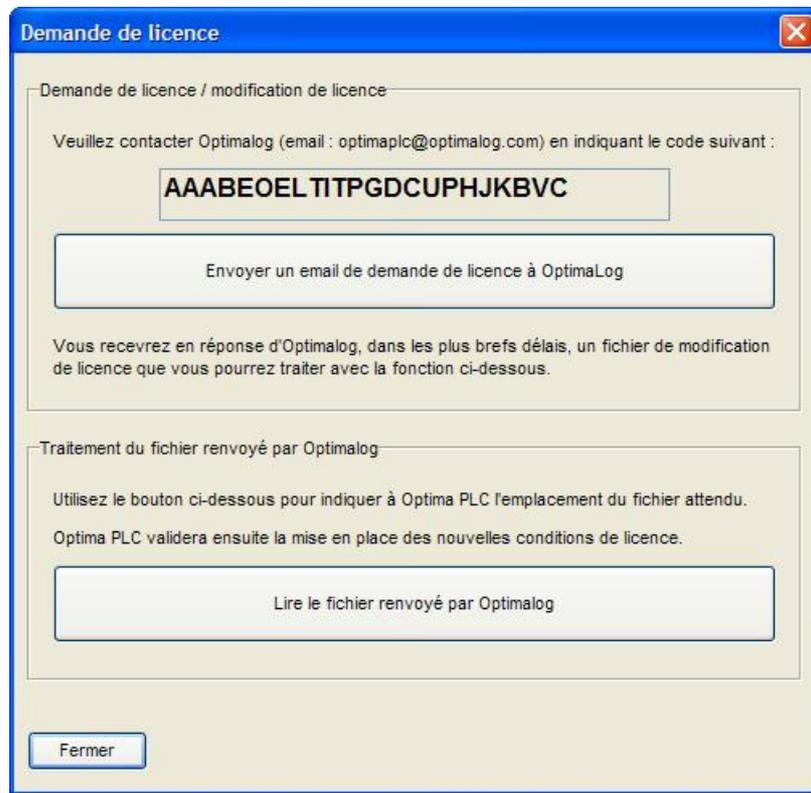
When a new dongle key is inserted, actualization is not instantaneous. To force it, click on the button "Update", the information of license are then read then since the key.

The displayed informations indicate the general features of the license (rights of use of the workshop, of the runtime...) and the dates of deadlines if there is some. If no date is mentioned in a field, there is no time limit.

The software maintenance end date indicates that the license won't permit to make operate a version of the software of which the date of creation is beyond.

To prolong the limits of validity, thank you to contact your distributor or OptimaLog.

The button "Modification Licenses" allows access to a new screen :

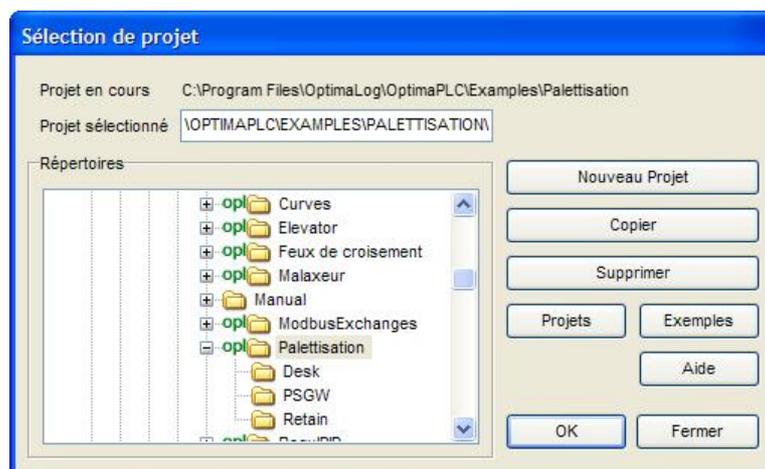


The mentioned code must be communicated to Optimalog, at the same time as the demand of license modification (new license, upgrade, extension of duration, etc). The code can be copied in the clipboard, or sent by email by clicking on the button 'Send an email...'.

When the demand is accepted and is treated by Optimalog, a file is sent to validate the modification. Click on the button "Read the file sent back by Optimalog" to select this file and to do the modification.

2.1.2 Menu Project

2.1.2.1 Project selection



The dialog box of "Project selection" is constituted of the following elements :

- The indicator of the project in progress (to the starting of Optima PLC, this field is empty)
- An input box to select the project (name hit by the user or put up to date at the time of a selection in the tree of the directories)
- A panel "Directories" to select an existing project in the tree of the disk or a directory to create a new project. The directories constituting valid projects Optima PLC are marked by an OPL symbol.

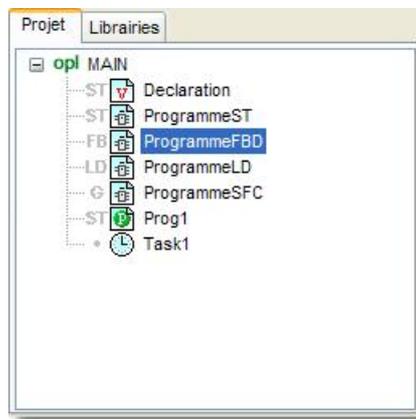
- A button "New project" to create a project : the directory in which it must be created must have been selected before clicking on this button, the name of the project must be composed of alphanumeric characters.
- A button "to Copy" to copy an existing project under another name, the project source is not suppressed at the end of the copy (remark: the copied libraries are not famous)
- A button "to Suppress" to suppress the selected project, two confirmations are necessary before the project is suppressed really
- Two buttons "Projects" and "Examples" to select the directories of project and examples created by default in the directory Optima PLC
- A "OK" button to validate the choice of the project (a double-click with the mouse produces an equivalent action)
- A button "Close" to close the window and to come back in progress to the project

When a project is selected, a new dialog box appears :

This window permits to define (and to visualize) the information relative to the project, as :

- The user's name
- The title of the project
- The type of project (choice between a project Apigraf PLC and a project Optima PLC)
- The number of version (composed of 3 indications major, minor and build, V0.0.1 at the creation)

2.1.2.2 List of the POUs



The editor is composed then of a menu, of an icons bar, of a tab panel to visualize as a tree the content of the libraries (on the left), of a second tab panel (on the right) to display and to modify the data of the POUs of the libraries and of a statusbar (underneath) for the results of compilation.

2.1.2.3 Editor Optima PLC

The editor appears like a simple editor of text files, the menu bar consists then of 2 supplementary options :

- File: to create, to open, to close or to save text files
- Find : to look for or to replace some words in the open file

2.1.2.4 File manager

Opening of the file explorer of Windows. The focus is placed on the directory of the current project.

If the project is an Apigraf project, the Apigraf file explorer is displayed.

2.1.2.5 Window system

A window system can be directly open from Optima PLC without passing by the menus of Windows. The directory of the project is selected automatically as current directory.



2.1.2.6 Quit

The development workshop Optima PLC only closes itself after having asked the user if he wishes to save the libraries under modification.

2.1.3 Menu Files

2.1.3.1 New library

This option creates a new library Optima PLC, placed after all already created those. The name of the library must be enter at this moment. The workshop creates then an empty file .opl with this name.

2.1.3.2 Open a library

Option permitting to open an already existing library on the disk and to integrate it in the project.

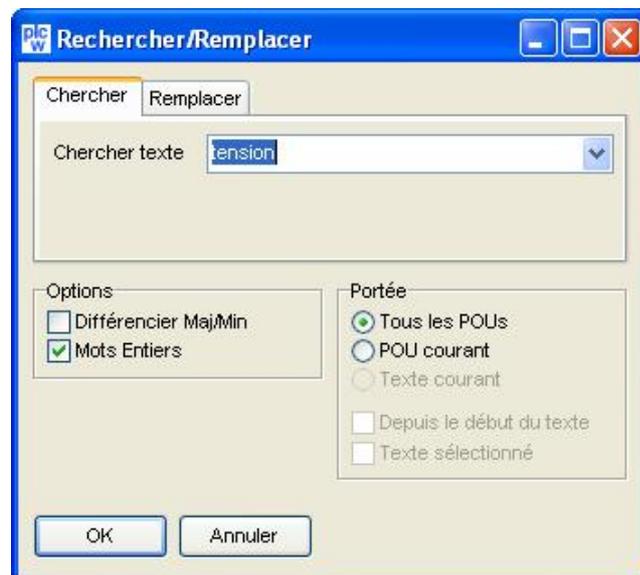
2.1.3.3 Save all libraries

Save all modifications done in all open libraries.

2.1.4 Menu Edition

2.1.4.1 Find

This option permits to look for a text in the project.



2.1.4.2 Replace

This option permits to replace a text by another in the project.



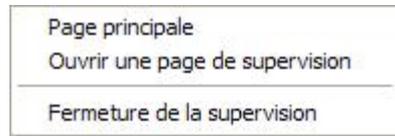
2.1.4.3 Next Occurrence

When a text is searched or to be replaced, this option permits to select the following occurrence.

The F3 function key also permits to do this operation.

2.1.5 Menu Supervisory

This menu allows the access to the workshop supervisory integrated into Optima PLC.



- Main form

This option calls the supervisory workshop by displaying the main form. If this form doesn't yet exist, it is created.

The main form corresponds to the file situated in the PSGW subdirectory, and whose name is the one of the project. The extension of the supervisory forms is .PSGW.

- Open a supervisory form

This option permits to choose a supervisory form to edit.

- Closing of the supervisory

This option provokes the closing of the supervisory workshop.

2.1.6 Menu Compilation

This menu includes the different options permitting to define the destination of the application, to compile it and to execute it.

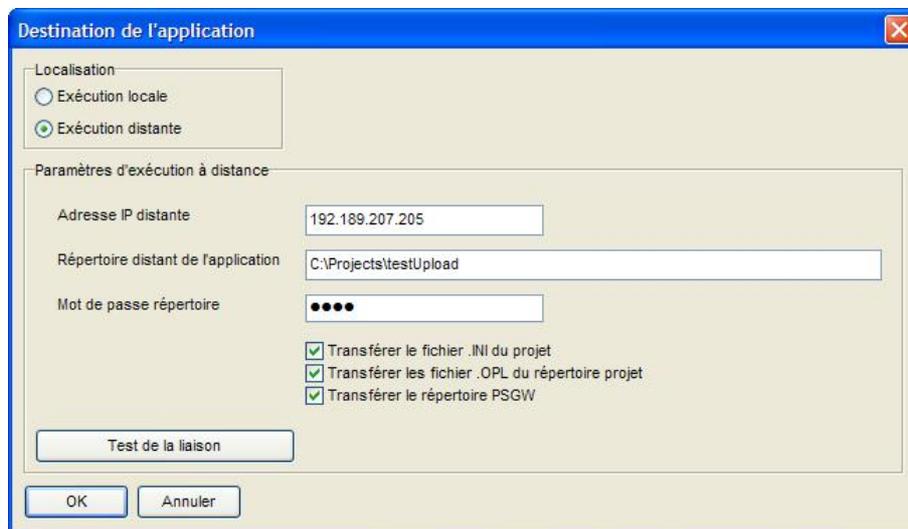


2.1.6.1 Destination

This option permits to define the final site of the application. This site can be :

- local: the application is generated and is executed on the same PC that the development workshop.
- distant: the application is generated locally, then transmitted to the definite address.

In both modes, the application can be executed, stopped and modified on line since the workshop.



2.1.6.1.1 Conditions of working for a distant destination

To enable the remote execution function, the following conditions must be filled.

- Optima PLC must be installed on the distant station, with the same version that on the local station.
- Optima APIFTP Servers must be running on the distant station. This program is normally installed at the same time as Optima PLC.
- The configuration of the sharings in Optima APIFTP Servers must allow a complete access to the directory of the binaries of Optima PLC, as well as to directory intended to contain the application. Besides, the passwords must be identical for both directories.

In the case of a standard installation, the directories are defined as follows :

C:\Program Files\OptimaLog\Optima PLC\bin	directory of the executable (binaries)
C:\Documents and Settings\All Users\Documents\Optima PLC\Projects	directory of application

If the installation has been made in non administrator session, the directories are :

C:\Documents Settings and \ [user] \Application Data\OptimaLog\Optima PLC\bin	directory of the executable
C:\Documents Settings and \ [user] \My documents	directory of application

\Optima PLC\Projects

- If the distant PC is protected by a firewall, it must allow the TCP ports 10260 (APIFTP) and 10580 (communication between workshop and runtime).

The window of parameterization includes a button "test" permitting to verify if the execution is from afar possible. This button doesn't test the link necessary to the online modification.

2.1.6.1.2 Parameters for a distant destination

- Distant IP Address

Address of the PC destination. This address can be local or accessible via Internet.

- Distant directory of the application

Access path to the directory in which the files of the application must be copied.

- Password directory

Definite password in the sharings on the distant PC with Optima APIFTP Servers.

- Transfers

Definition of the files transferred at the transmission of the distant application. These files are not transferred at the time of an online modification.

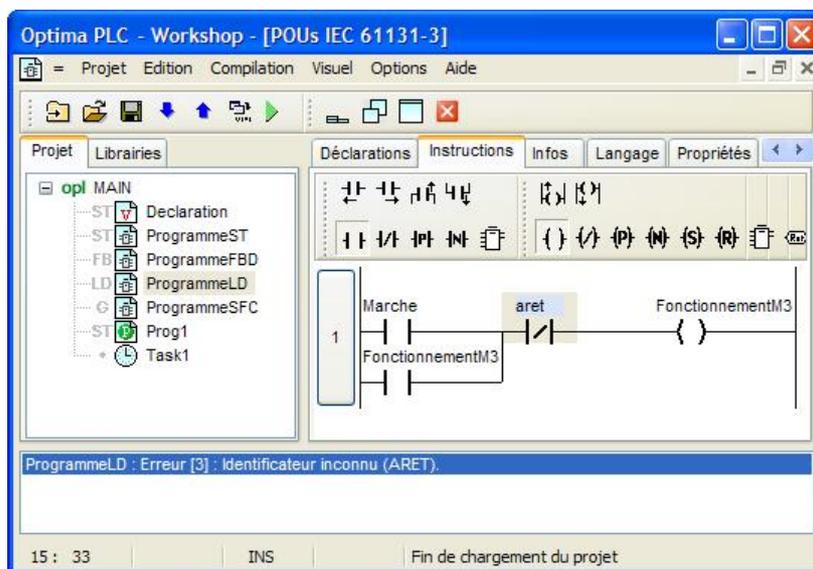
Attention: the transfer of the .ini file is necessary at the time of the first execution, but then crushes the written information locally by the application.

2.1.6.2 Full compilation

The full compilation recompile all POU's and files of the project. If an error of compilation occurs, the compilation is stopped on the corresponding position.

In case of error, the indications necessary to the debug are displayed in the display at the bottom of the editor.

Below, in the example the variable "Stop" has been hit with an error (a r misses). The error is signalled on the line of the low. A double click on this line sends back directly to the error position.



2.1.6.3 Compilation and online modification

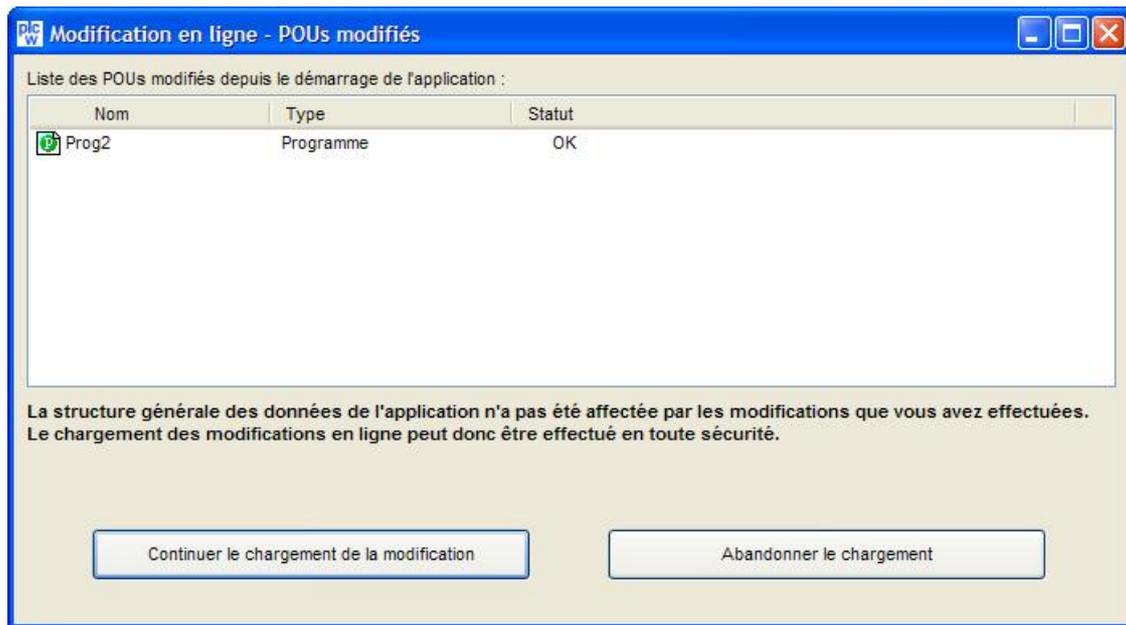
This option permits to compile the application whereas this one is already running.

Firstly a confirmation is asked this stage :



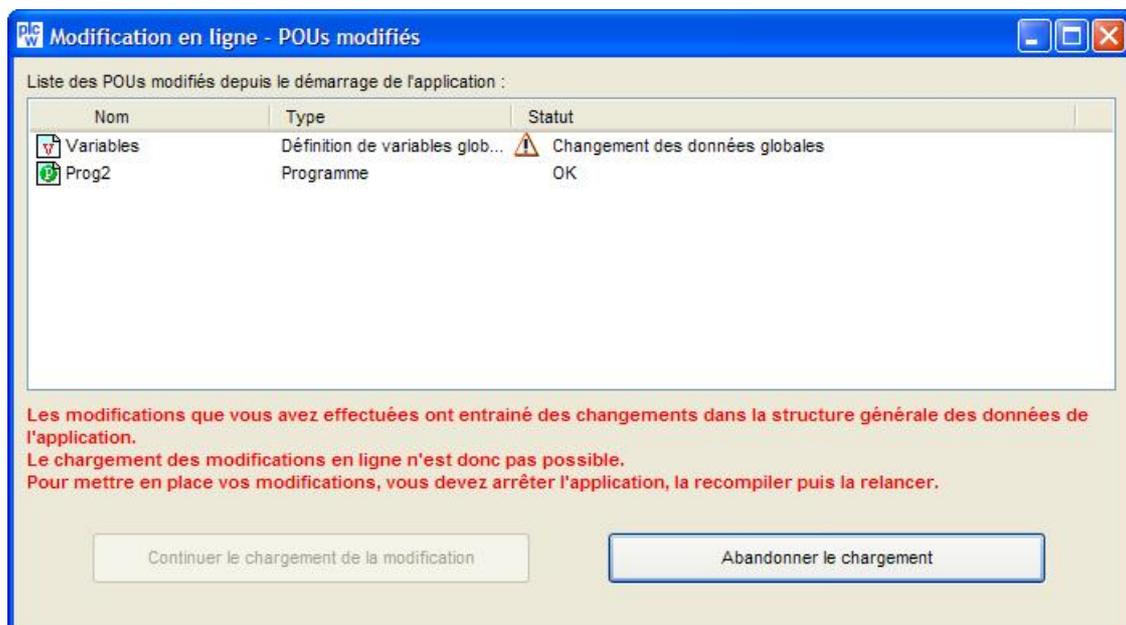
If this confirmation is given, Optima PLC verifies the consistency of the POU's as well as the modifications brought to the application since its launching.

If all is correct, a window as this one is displayed :



In this case, a click on "continue the loading of the modification" provokes the generation of the executable code, and transmits the demand of loading of this new code to the application. This one that takes in charge the setting up of the code modified, after the user's last modification.

On the other hand if some non authorized modifications are detected, a window as this one is displayed :



In this case the on line loading of the modifications cannot be done.

A complete description of the authorized modifications or no is available to the chapter "[On line modifications](#)"^[376].

2.1.6.4 Execution

In the case of a project Optima PLC, if a modification has been done in a part of the project, the application is first of all recompiled, then executed locally.

The local execution is not possible in the case of an Apigraf project.

2.1.7 Windows menu

2.1.7.1 Next document

This option is only available with the editor of text files of Optima PLC (and no the list of the POU's).

When several files are open, it is possible to put in foreground the following document.

2.1.7.2 Previous document

This option is only available with the editor of text files of Optima PLC (and no the list of the POU's).

When several files are open, it is possible to put in foreground the previous document.

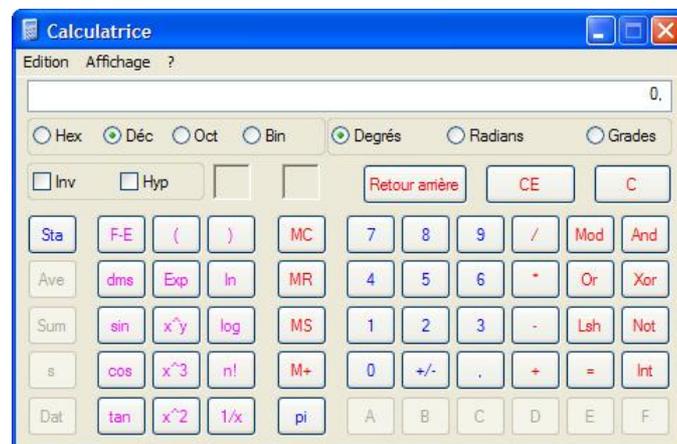
2.1.7.3 Video mode

It is possible to reach the display properties of Windows quickly from Optima PLC.



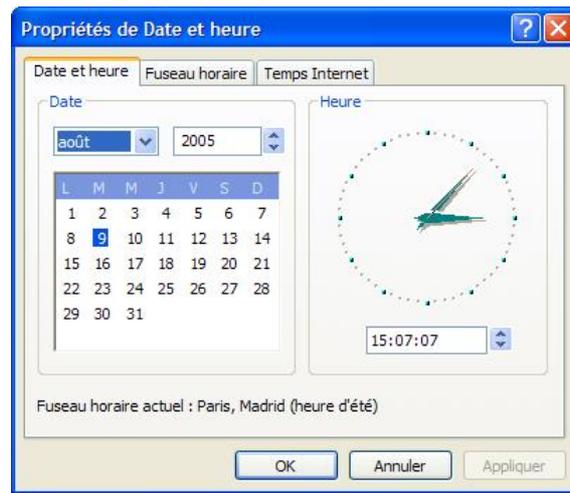
2.1.7.4 Calculator

It is possible to reach the calculator of Windows quickly from Optima PLC.



2.1.7.5 Calendar / Clock

It is possible to reach the date and hour properties of Windows from Optima PLC.



2.1.7.6 Format Dates / Hour

It is possible to reach the regional and linguistic options of Windows quickly from Optima PLC.



2.1.7.7 List of the windows

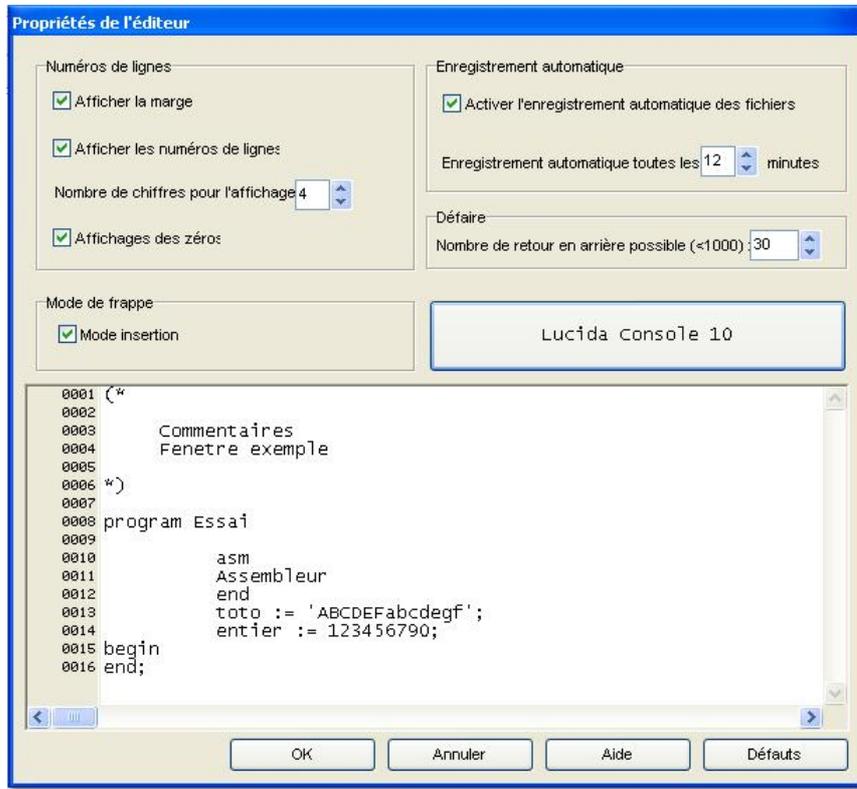
This option is only available with the files editor of Optima PLC (and no the list of the POU's).

The list permits to know quickly what are the open files but also to put in foreground the file that one selects in the list.

2.1.7.8 Editor Options

2.1.7.8.1 General options

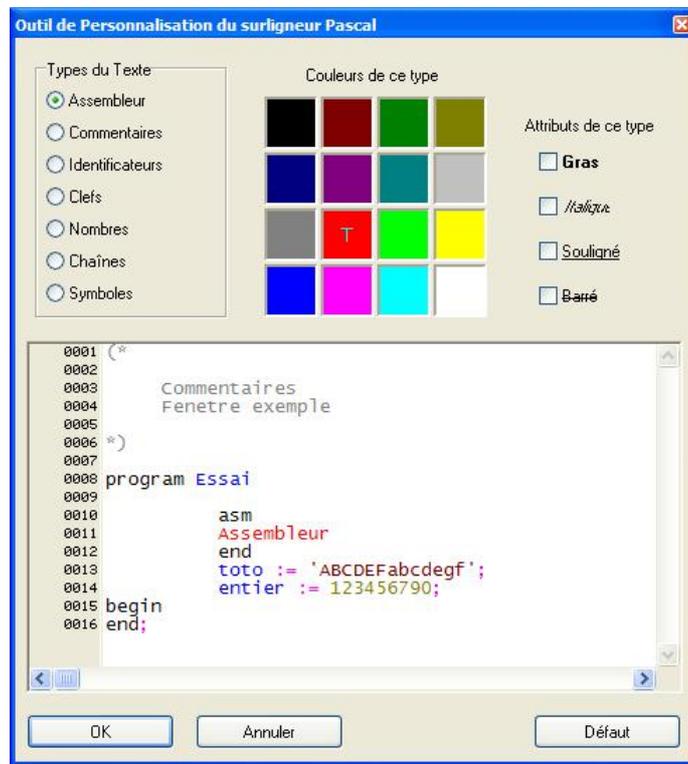
- Numbers of lines: it is possible to display the margin and the number of the lines (with different formats).
- Automatic backup : the backup can be done in an automatic way, in this case it is possible to specify the period.
- Undo: the annulment of the last strikings is evidently possible, the number of possible annulments is configurable
- Striking mode : the striking can be made in insertion mode or no
- Font : a button indicates the name of the font used, by clicking over it is possible to modify this font
- Window example : it is possible to see in real time the aspect that the editor will take with the selected options



2.1.7.8.2 Pascal editor

The dialog box "Pascal highlighter option tool" permits to specify for every type of text what is the color and what are the font attributes that it is necessary to use to put in evidence this type of text.

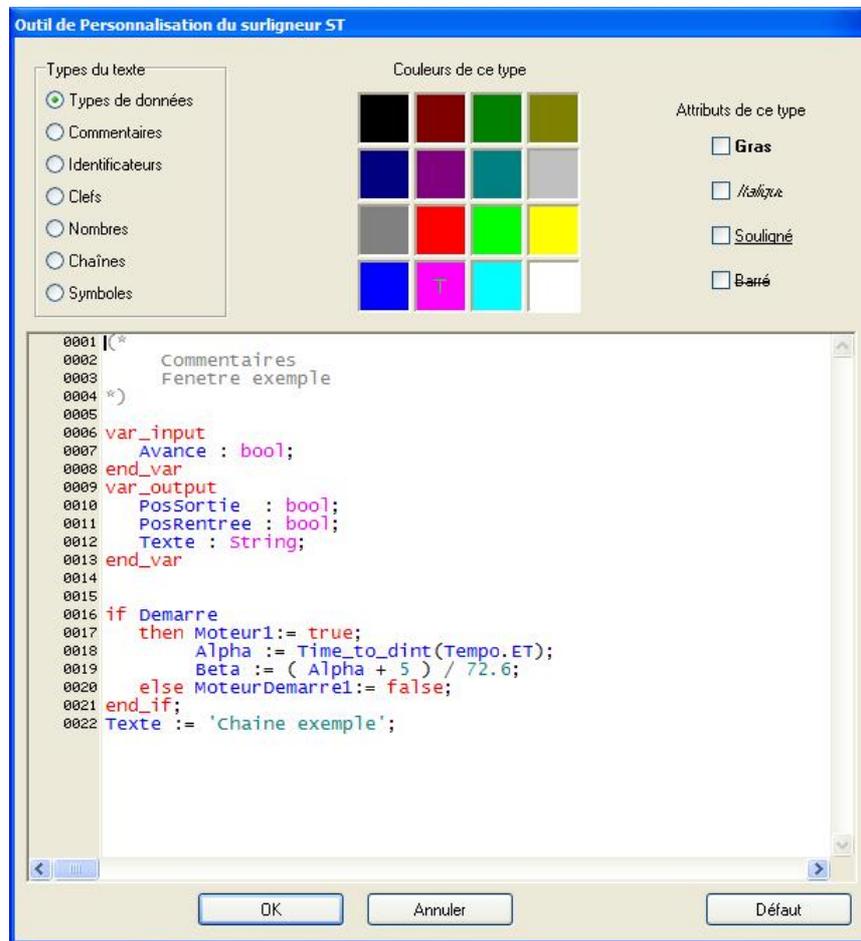
An example window permits to see in real time the aspect that the text will take with the selected options.



2.1.7.8.3 ST editor

The dialog box "ST highlighter option tool" permits to specify for every type of text what is the color and what are the font attributes that it is necessary to use to put in evidence this type of text.

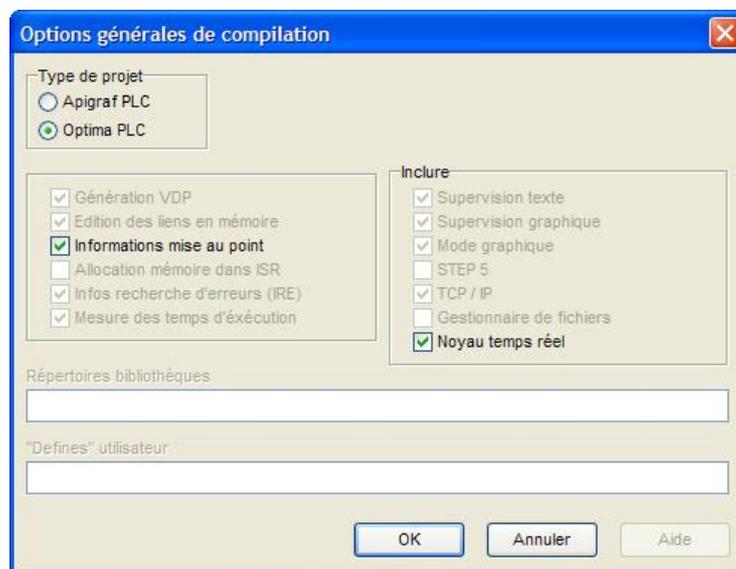
An example window permits to see in real time the aspect that the text will take with the selected options.



2.1.8 Menu Options

2.1.8.1 General compilation

In a project Optima PLC, the following options are available :



- Debug information

When this option is checked, the compiler generates the supplementary verification code to detect possible limits errors, in particular on the arrays indexes. The program is therefore slightly slower, however these verifications can detect some problems of programming quickly. Its use in debug phase is recommended strongly.

- Real time kernel

This option permits to deactivate the kernel real time integrated in Optima PLC. The application can continue to operate if this kernel is deactivated.

2.1.8.2 System constants

It is possible to define the maximum number of PLC variables usable in the program.

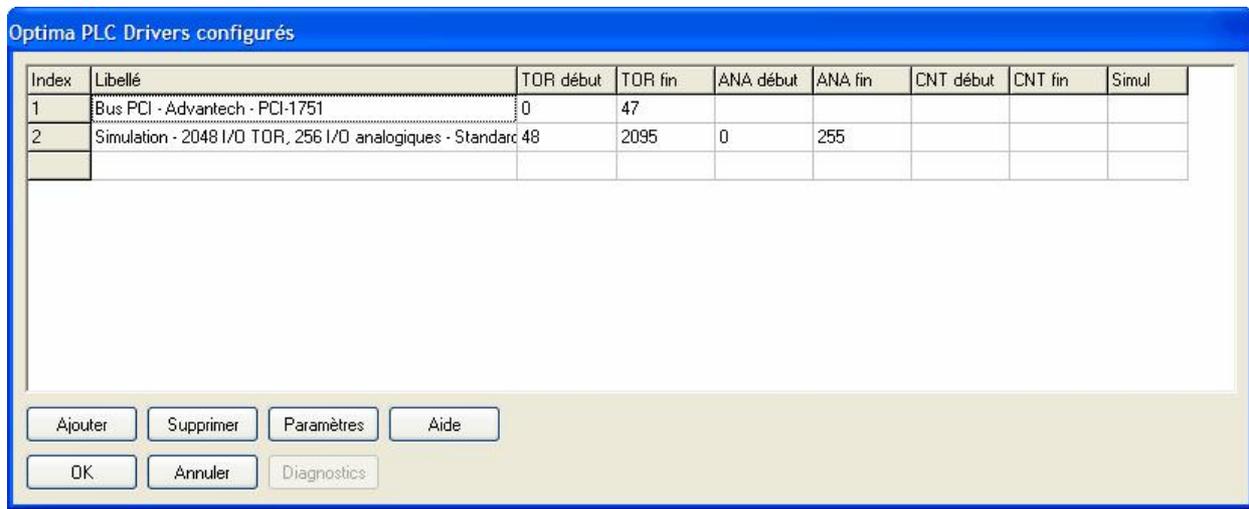
Paramètre	Valeur
NoMaxR	499
NoMaxRR	499
NoMaxL	499
NoMaxG	199
NoMaxSA	499
NoMaxC	99
NoMaxA	199
NoMaxDT	199
InitNbInterSeq	1024
Base de temps	0.001
Taille pile	65520
Taille pile seq	16384
Adresse carte RAM	0

2.1.8.3 Settings editor

The settings editor permits to visualize the data contained in the initialization file of the project.

2.1.8.4 Hardware interfaces

In a project Optima PLC, the definite drivers allow to manage the inputs and outputs, digital as well as analogical.



The dialog box permits to add, to suppress or to configure the hardware interfaces. The interfaces can also be put in simulation.

The button "to Add" permits the display of the list of the available hardware interfaces. When the choice is validated, the dialog box of parameterization is immediately displayed ; some parameters are obligatory to create the interface (examples: board address, IP address).

When a hardware interface is added, numbers of input/outputs and channels are automatically reserved. The range of numbers is displayed in the list of the interfaces, however to know the affectation of the numbers to the inputs, to the outputs, to the input channels and to output channels, it is necessary to consult help associated to the selected interface.

2.1.8.5 Text output files

This option permits to define the accesses to the output text files used in the application. This term regroups the outputs as well on disk real that the outputs on parallel printers (LPT1, LPT2, etc.). Here, only the outputs on file disk are concerned.

The outputs are done "with the current", that is to say that the written data are added following the previous.

Only the outputs on disk can be configured here. With this type of access, the files grow to every new writing of data. These data must be reset therefore, for example while using the functions clean to the files: RenameFile (to reappoint the file), DeleteFile (deletion of the file).

In the PLC program, the data can be written in the files thanks to the TOWrite functions and TOWriteLns (idem TOWrite with return to the line).

Example :

```

if SA10
  then TOWriteLn( Essai, R1, '$T', R2 );
        SA10:= false;
end_if;

```

When the SA10 flag passes to 1, the value of the R1 registers and R2, separated by a tabulation, are written in the file whose name interfaces is "Test." A return to the following line is automatically achieved.

The box of configuration dialog permits to parameterize the output text files used (on disk) in the application.

Coefficient of units conversion

This coefficient permits to correct the values stored in the table for display. For example, if the values are memorized in milliseconds, but if the displays must be made in seconds, then the units are seconds, and the coefficient of conversion is 0.001.

Number of points

This parameter indicates the size of the table. Every value capable to be stored in the table occupies 4 bytes (integer 32 bits).

Point Type

This parameter is used for the display in the curves, in the case where the table acts as abscissa. The possible values are the following :

Normal : the numeric values are displayed as is.

SecondFactor : the numeric values correspond to absolute times in second. In this case, the displayed values are the hour and the date correspondents.

SecondFactor without date : same thing that the previous option, but the date is not displayed (hour solely).

- **Add**

This order permits to add a new table in the list. The box of parameterization appears (see above), permitting to define the parameters of the table.

- **Delete**

This order suppresses, after a confirmation, the table selected in the list.

2.1.8.7 Miscellaneous**2.1.8.7.1 Startup settings**

The options of configuration of the starting concern the choice of the project and the opening of files in the editor Optima PLC.

2.1.8.7.2 Users management

This option permits to choose the users management mode, in which an identification is asked previously to the workshop screen access. This mode offers a better security in an multi-users environment, as on a production machine.

Attention : it is necessary to not confound the users management of Optima PLC, proposed here, with the one of Windows. Windows permits to define the different users having each the possibility to open a session, with different rights.

When this mode is activated, the rights attached to the user restrict him the access to some functions and menus (once again, it is about the rights for Optima PLC, and not for the Windows users). Besides, a special menu of users management is displayed.

2.1.8.7.3 Windows session for execution

This option permits to open a dialog box with which the Windows session used for the execution of the application can be parameterized.



The parameters correspond to the user login whose rights must be used to execute the application. These parameters are memorized in the file of configuration of the application (the password is strongly encoded).

This configuration is not necessary if the session used to execute the application is an administrator session. Administrator's rights are indeed indispensable to allow the access to some resources of the PC.

A non administrator session is counseled however to limit the risks of access to non authorized data, of destruction of data, of configuration corruption, etc... In this case, the application must function in an administrator session configured here.

The button "Test" permits to verify the parameters. If the configured session is correct, the following window is displayed :



2.1.8.7.4 Windows sessions settings

This option gives access to the function of parameterization of the sessions of Windows.



This dialog box is only available if Optima PLC is currently executed from an administrator session.

To define the session with which Windows automatically starts by default, uncheck the box "The users must enter a name of user and a password to use this computer."

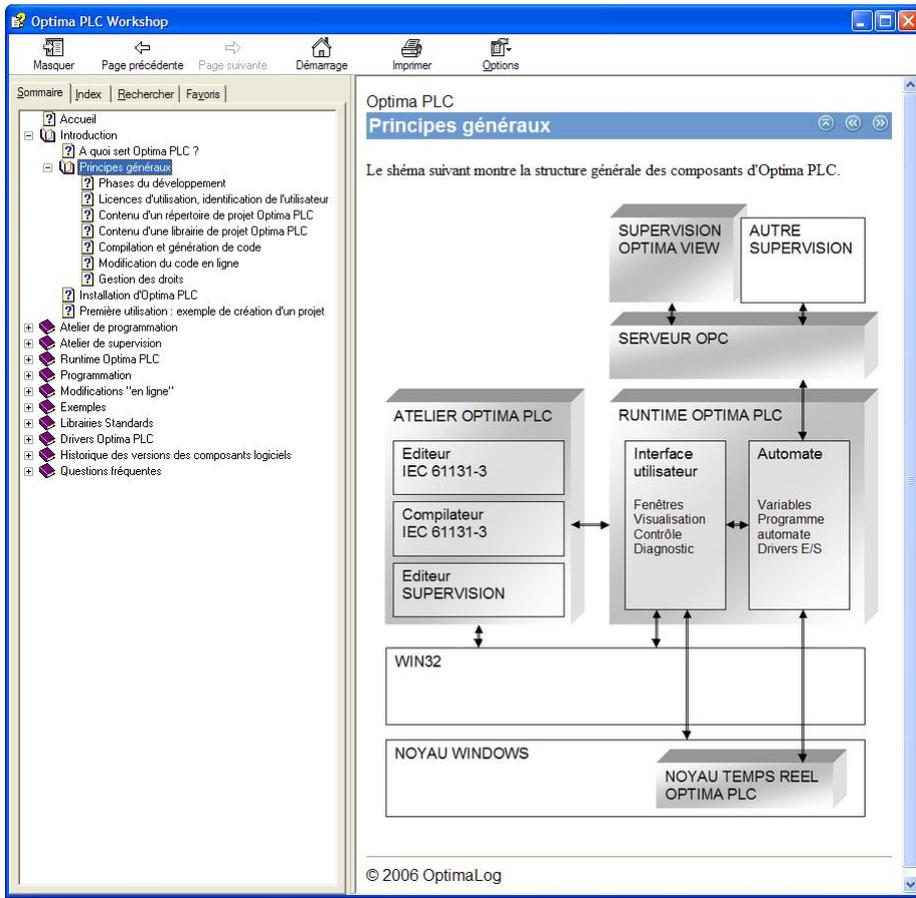
2.1.8.7.5 Default Files

It is possible to define the type of file proposed by default in the demand of file opening in the editor Optima PLC.

2.1.9 Menu Help

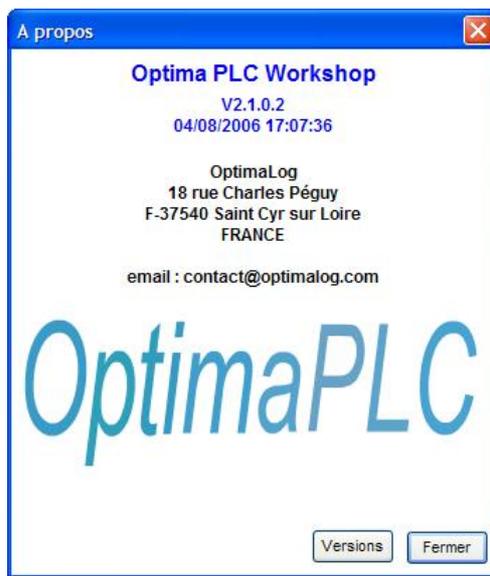
2.1.9.1 Help

It is possible to visualize this help file from the software package Optima PLC.



2.1.9.2 About Optima PLC Workshop

This option permits to get the information of version of the software.



The button Versions provokes the display of the list of the software components and their version.

Programme	Version	Date
APIEXPLORER.EXE		06/09/2002 12:19:58
APIFTP.DLL		24/10/2003 17:48:36
APIGRAFC.EXE		04/11/2005 16:16:24
APINVIEWER.EXE	1.1.0.5	02/08/2006 15:43:02
BORLNDMM.DLL	5.0.6.18	25/02/2000 06:01:00
CNRMC416.DLL		05/11/2001 03:40:00
COPYAPS.EXE		08/04/2004 12:00:44
DBGMSGCFG.DLL	1.0.0.1	20/06/2002 16:08:04
DEVICETREE.EXE	2.4	21/07/2001 20:31:14
DRIVERS.DLL		02/08/2006 14:49:58
DRIVERS.EXE	2.1	02/08/2006 14:49:30
EDITOR.EXE	2.1.0.2	04/08/2006 17:07:36
ESTEEMPROCS.EXE		29/09/2004 23:14:14
IEC1131.DLL	2.1	02/08/2006 15:09:24
IO.DLL		23/09/2005 15:01:20
OPTIMAPLCRUNTIME.EXE	2.1.0.10	03/08/2006 18:10:14
PSGDLL.DLL		03/05/2004 15:20:24

Fermer Historique des versions

2.2 Description of the icon bar

2.2.1 Tool bar of project management

	new library : it is possible to add a library in a project. The library is created and added following the existing libraries. The name of the new library must be entered at this moment, and the corresponding empty file .opl is created on the disk.
	open a library : select a file with the .opl extension, the corresponding library is integrated then to the project.
	save : the modified libraries are stored on disk. The F2 key has the same effect.
	Move down : the selected POU is displaced downwards in the tree of the library
	Move up : the selected POU is displaced upwards in the tree of the library
	compile : a full compilation of all POUs is done, in order to generate the executable code. If the application is running, this icon permits to activate the on line modification .
	execute : the program, currently stopped, is compiled if it is not already, then is executed.
	stop : the program currently running is stopped after a confirmation.

2.2.2 Tool bar of windows display management

These options concern the edit windows of the POUs opened in the right zone of the main window.

	permits to reduce all windows
	the windows are normally displayed with their size and position
	all windows are maximized
	all windows are closed

2.3 Use of the libraries

2.3.1 General principles of the libraries

A library Optima PLC is a file containing the elements of programming (POUs). The extension of this type of file is .OPL (Optima PLC Library).

A project may be constituted of several libraries, these libraries permit to get a good modularity of programming,

by regrouping by theme the POU's in specific libraries. For example, the standards elements of programming provided with Optima PLC (several hundreds of POU's) are regrouped in specialized libraries.

2.3.1.1 Content of a library

A library is constituted of POU's of different types :

	task ³²	Schedule the execution of the programs.
	global variables ³³	Declarations of available variables for all the programming POU's.
	types ³³	Declarations of type of available variables for all the programming POU's.
	block of program ³⁴	Main programming.
	function block ³⁴	Subroutine, with or without return value, that memorizes the state of its internal variables between 2 calls. So for a same set of input parameters the result returned by this block can be different (example of the PID: for a same order the output of this regulation block evolves until it reaches the order).
	function ³⁴	subroutine whose return value remains identical for a same set of input (no memorization of the variables internally) parameters.
	Action	subroutine used in the SFC diagrams, permitting to describe an action to do from the steps.
	subgroup ³⁴	POU of library structuring. Its unique object is to contain other POU's.

2.3.1.1.1 Task

In accordance with the norm IEC 61131-3, a task is a POU which provokes the execution of programs or function blocks periodically, or in answer to the change of state of a boolean variable.

The parameters associated to a task are :

- its name
- the level of priority for the execution of the programs or function blocks

The levels of priority of the tasks are defined from 0 to 9 as follows :

Level of priority	Use	Time real
0	Maximal priority, used internally by Optima PLC.	yes
1		yes
2		yes
3		yes
4	"Standard" priority level for the PLC program. The actualization of the tables of the inputs/outputs is done to this level of priority, that must be used therefore for a "synchronous" working with these updates.	yes
5		yes
6		yes
7	Minimal real time priority	yes
8	Level of priority non real time, to use for the tasks doing brief calls to the API of Windows and having to be executed to an elevated frequency (network,...).	no

9	Level of priority non real time, to use for the tasks doing non priority calls to the API of Windows (files,...).	no
----------	---	----

- the trigger (that is the name of the optional boolean variable provoking the execution).

This variable, if it is defined, must be of BOOL type, and to be accessible to the global level. In this case, the task executes itself at a rising edge of this variable value.

- the period of execution of the task

The period must be defined with the syntax IEC 61131-3, that is to say with the T# prefix. For example: T#5ms (5 milliseconds), T#4s (4 seconds), etc.

Possibly, the period can be specified with a global variable of TIME type.

If nor the period nor the trigger are defined, the task is executed to the maximal frequency allowed by the system (currently 1000 Hz).

- the list of the programs or function blocks to manage

This list understands, for every program, the name of variable, the type of program block, and the possible list of the parameters.

- the phases of execution of the runtime during which the task is allowed.

The runtime passes by several phases of execution. The tasks defined by the user can be allowed or not to run during some among them :

Phases	Interpretation
START PLC	Task of initialization, permitting to do some operations only at the starting of the application. When the task finished its initialization, it must set to 1 the boolean variable StartPLCPhaseOK
RUN	Phase of normal working of the runtime.
FAULT	Phase of working damaged, a subsystem at least is not operational.
STOP PLC	Task of finalization, permitting to do some operations only at the end of the execution of the application. When the task finished its execution, it must set to 1 the boolean variable imperatively StopPLCPhaseOK

In most cases, the classic tasks are allowed to function in RUN phase and FAULT.

2.3.1.1.2 Global variables

One "Global variables" POU permits to declare the variables that will be accessible for all POU's of programming (declared after in the project).

As this POU is dedicated to the declaration of variables, the framing of the declarations with the keywords VAR and END_VAR is automatically inserted at the creation of the POU. However, other declarations can be added, with VAR CONSTANT or VAR RETAIN. The POU can contain several blocks VAR ... VAR.

The variables can be the PLC variables or inputs / outputs, in this case the declaration doesn't have for goal "to create" the variable but merely to affect a mnemonic and/or a commentary.

2.3.1.1.3 Types

A "Types" POU permits to declare types of variables that will be usable by all POU's of programming (declared after in project).

As this POU is dedicated to the definition of types, the framing of the declarations with the keywords TYPE and END_TYPE is automatically inserted at the creation of the POU.

2.3.1.1.4 Block of program

A block of program is a logical assembly logical of instructions and POU.

It is composed of a local declaration part and an implementation part.

The declared variables can be input variables, of output, local or temporary, the framing of these different declarations with the keywords of the norm IEC 61131-3 is automatically inserted at the creation of the POU.

To the difference of the function blocks, the programs can contain the VAR_GLOBAL declarations, that will be used then in function blocks with the VAR_EXTERNAL declaration.

The implementation part can be done in any language available in Optima PLC.

2.3.1.1.5 Function block

A function block is a subroutine with internal remanent variables

As the block of program, it is composed of a local declaration part and an implementation part.

The declared variables can be input variables, of output, local or temporary, the framing of these different declarations with the keywords of the norm IEC 61131-3 is automatically inserted at the creation of the POU.

The implementation part can be done in any language available in Optima PLC.

2.3.1.1.6 Function

A function is a subroutine whose possibilities are least that those of a function block since the internal variables are not memorized between two calls and only one output variable is allowed.

The declared variables can be input variables or temporary, the framing of these different declarations with the keywords of the norm IEC 61131-3 is automatically inserted at the creation of the POU.

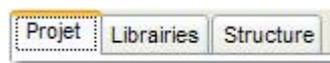
The implementation part can be done in any language available in Optima PLC but in SFC diagram.

2.3.1.1.7 Subgroup

The subgroup is a structuring element, it includes neither declaration nor instruction. Its goal is to group some POU's so that the library is organized in a logical way.

2.3.1.2 Trees of the POU's

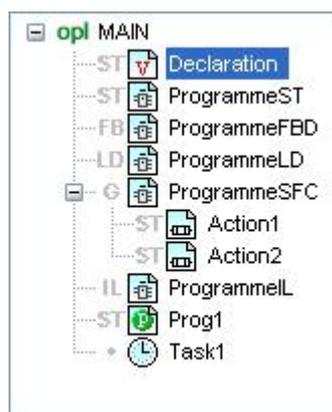
The left part of the screen of Optima PLC permits to visualize the POU's in tree views. The choice is made with the following tab :



2.3.1.2.1 "Project" conception tree

The composition of a library is displayed under a tree view (tab panel on left, tab "project").

The library tree constitutes **the conception tree**. It doesn't reflect the relations of dependence and ownership that exist between the blocks, contrary to **the execution tree displayed in the tab Structures and in the runtime**, but only the structure defined by the programmer for access convenience.



Only exception, the actions are displayed in direct dependence of the block to which they belong, and this relation is identical in the conception tree and in the execution tree.

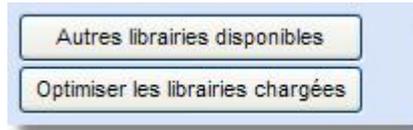
The "project" conception tree can contain several libraries. It is normally about the libraries specifically conceived for the project and situated in the directory of this project.

2.3.1.2.2 "Libraries" conception tree

This tree, similar to "project" tree, is available in the tab "libraries". It permits to reach the standard libraries of Optima PLC specifically, that is to say those that are installed at the same time as the product, or that the user created and placed in the OPL directory, in order to facilitate the sharing of it between several application.

The libraries of this tree are all accessible from the libraries of the tab project.

The following buttons are available in this tab :



- Other available libraries

This button permits to display the list of the available standard libraries, but not currently loaded in the project.

- Optimize the loaded libraries

When some libraries are loaded without being used in the project, they lengthen the compilation time uselessly. This button permits to unload the unused standard libraries automatically from the project.

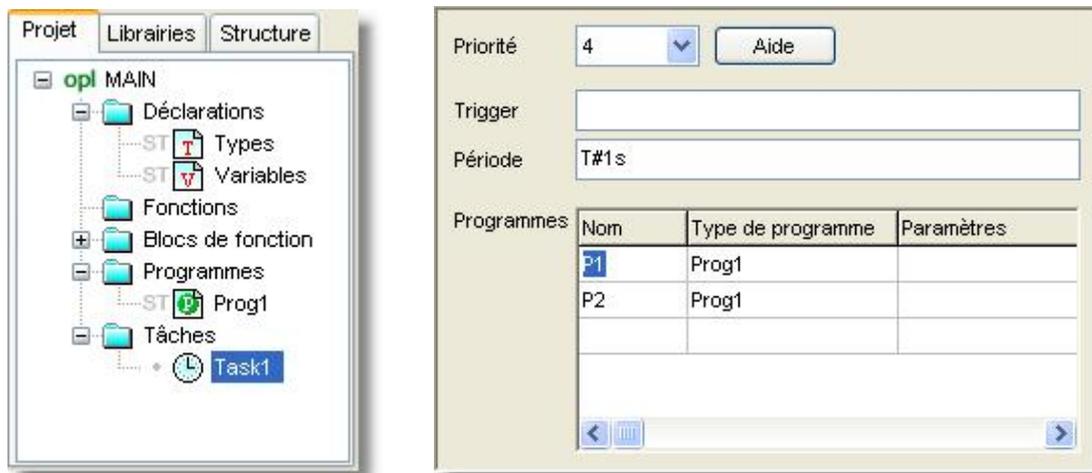
2.3.1.2.3 Structure (execution tree)

The structure of the application POU's corresponds to the overlapping defined by the declarations made in every POU, and therefore to the final structure of the application data. It is shown in a tree view in the "Structures" tab.

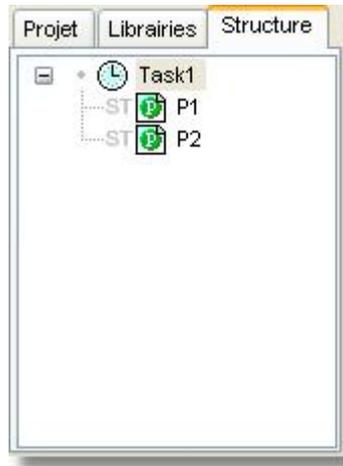
The structure begin at every defined task, that contains some programs, containing blocks of functions, etc... This structure is also called in Optima PLC "execution" tree. It can also be visualized in the runtime.

In this structure, the names of the instances of POU's (that is to say the variables) are displayed, and not the names of the types of POU's. On the other hand, the libraries and subgroups don't appear here (they serve to arrange the available POU's efficiently and don't have an impact on the structure).

Here is a project containing for example a task and a program. The task calls the program 2 times, under the names of P1 variables and P2.



The resulting structure is the following :



In this view of the structure, no operation of modification of this one is possible in this version of Optima PLC. However the selection of an element of the tree provokes the display in the right part of the screen, of the corresponding POU data.

The modifications done in the declarations of the POU's provoke the update of the structure if there is no compiling error. In case of error, after correction, use the function of application build to update the structures.

Option to Display all references

This option displayed at the bottom of tab allows to display, in addition to the structure of the blocks and programs, all the references to the functions and variable definitions.

2.3.1.3 "Project" Libraries

A library can be defined like being a "project" library. It is about a distinction of organization, in order to mark the really specific libraries to a project, and to distinguish them of those of general use.

A tab permits to choose between the display of the "project" libraries and "standard" libraries.

The "project" libraries are always placed after the other in the [conception tree](#)^[34], what allows them to make reference to all POU's placed in the standard libraries.

2.3.1.4 Definition order of the POU's

The definition order of the POU's in the library is very important : in Optima PLC, one POU cannot make reference that to the POU's that are placed above it in conception tree. For this reason, the declarations of types and global variables generally precede the definitions of blocks. The tasks are generally defined in last.

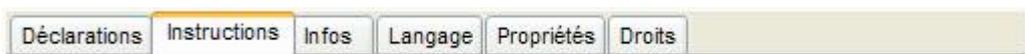
A POU can make reference to the POU's placed in other libraries, provided that these are placed before in the conception tree.

The libraries of the tab "Project" are logically placed after those of the tab "Libraries."

2.3.1.5 Properties of the POU's

For every POU, it is possible to visualize the data in the panel on the right, the number of available tabs depends on the type of POU as well as the user's rights.

Example for a block of functions :



2.3.1.5.1 Page Declarations

Contrary to the "global variable" and "Types" pages, it is indispensable in this page to frame the declarations by the keywords of the norm IEC 61131.

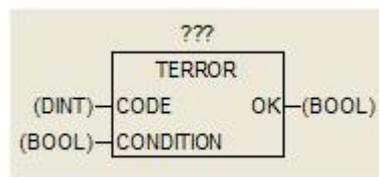
The keywords as well as the format of declarations to respect are described in the chapter "Programming" of this documentation.

```

0001 Var_Input
0002   Code                : dint;
0003   Condition           : bool;
0004 End_var
0005
0006 Var_Output
0007   OK                  : bool := false;
0008 End_var
0009
0010 Var
0011   Display             : bool := false;
0012   CodePrec            : dint  := 0;
0013   NewCodeDetected     : bool  := false;
0014
0015   // Sauvegarde de la demande d'affichage
0016   Displaying          : bool := false;
0017   // Message de confirmation pour l'acquiescement du défaut
0018   Msg                 : UserMsgErrorFB;
0019   ResultMsg          : dint;
0020
0021   // condition d'acquiescement utilisée pour l'affichage du message
0022   ConditionMsg       : bool;
0023
0024   IndexMsg           : dint;
0025
0026 End_var
0027
0028 Var_Temp
0029
0030 End_var

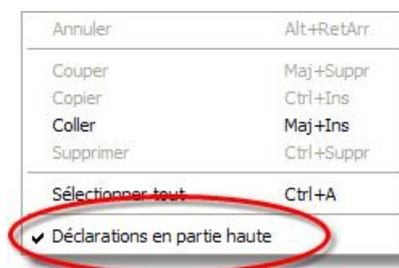
```

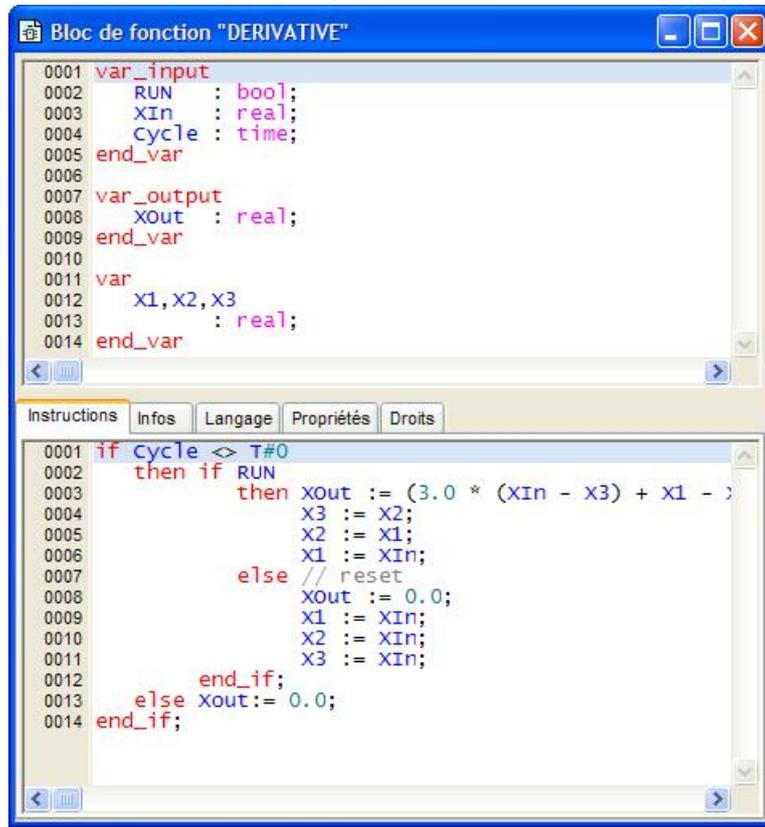
The declarations correspond above to a block of which here is the interface :



The variables "Code" and "Condition" are defined in input, the "OK" variable in output. The other variables defined in "var" are static variables used only internally in the block and that are not accessible from the outside.

The contextual menu of the declaration editor contains an option "Declarations in up screen part", permitting to take the declarations out of the tab and to display them in top of the window. The selected mode is kept for the ulterior displays.

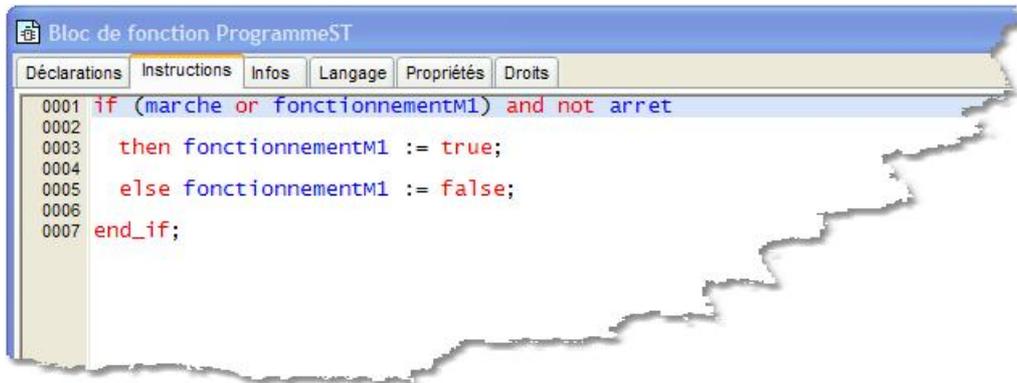




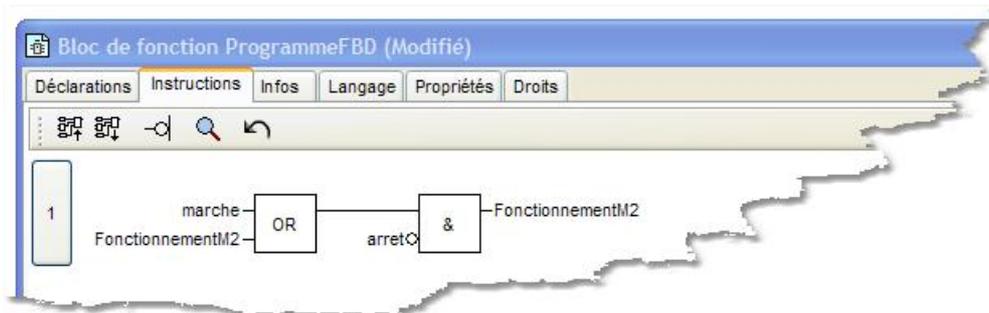
2.3.1.5.2 Page Instructions

This page takes a different aspect according to the type of language selected.

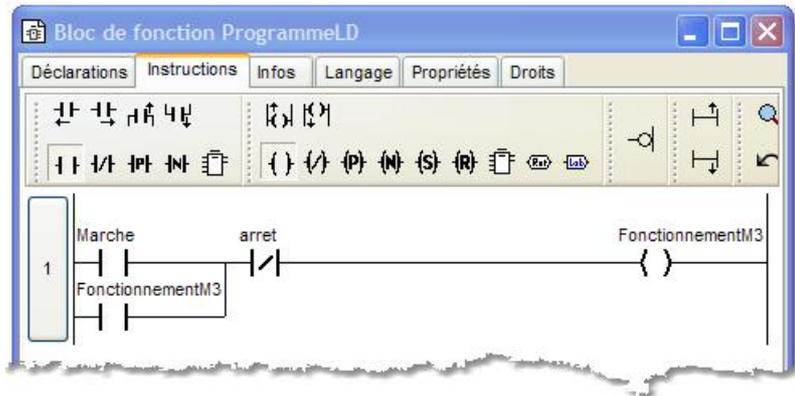
ST, IL and SFC text : the page behaves like a simple text editor. The syntax specific to the language is enhanced however.



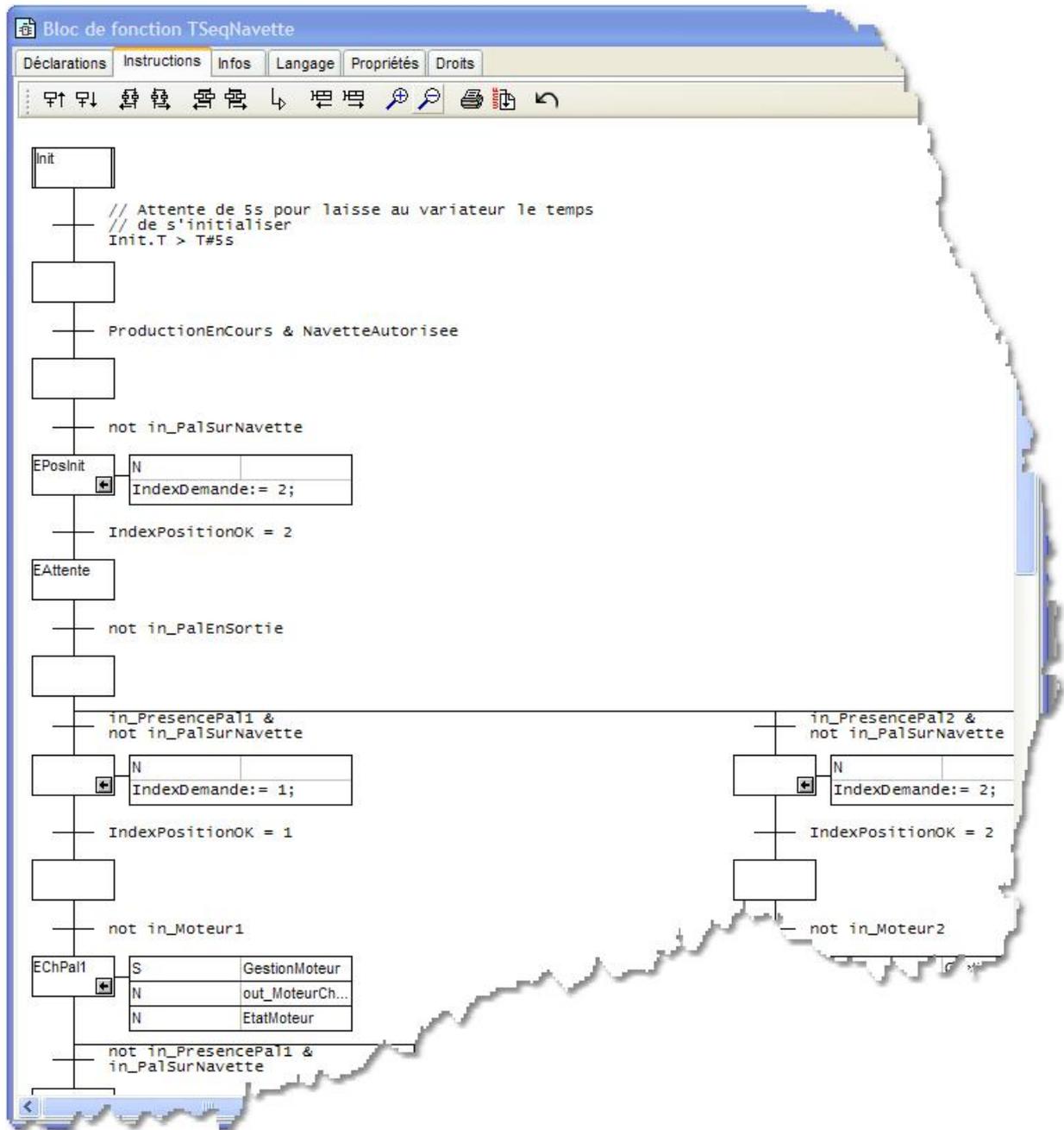
Diagrams of Block of functions, Ladder and SFC : the page is composed of a toolbar and of an edition zone for the diagram



Blocks of functions



Ladder

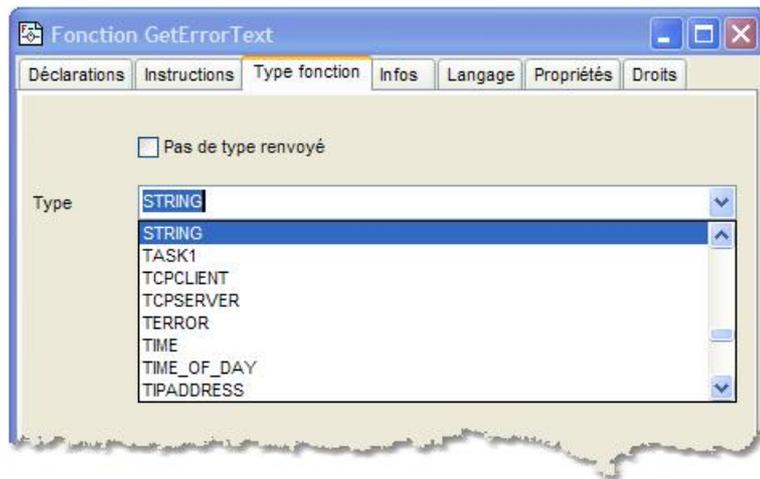


SFC or GRAFCET

The keywords as well as the format of the instructions are described in the chapter "Programming" of this documentation.

The use of the icons in the graphic languages is described in the chapter "[Creation of a library](#)"^[47].

2.3.1.5.3 Page type of function



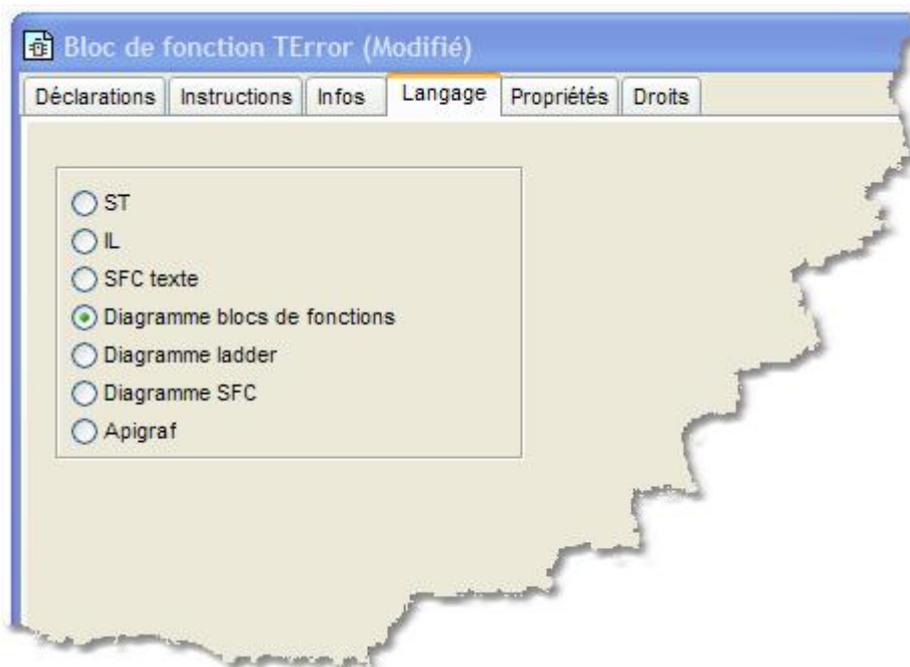
This page permits to define the type of the output variable of the function.

If the checkbox "No type returned" is checked, the function doesn't return any value, it is a "procedure".

In the contrary case, enter on the type input the name of the value type returned. The line permits to display the list of all types knew by the system.

2.3.1.5.4 Page Language

The page is composed of a limps to unique choice among the propositions.



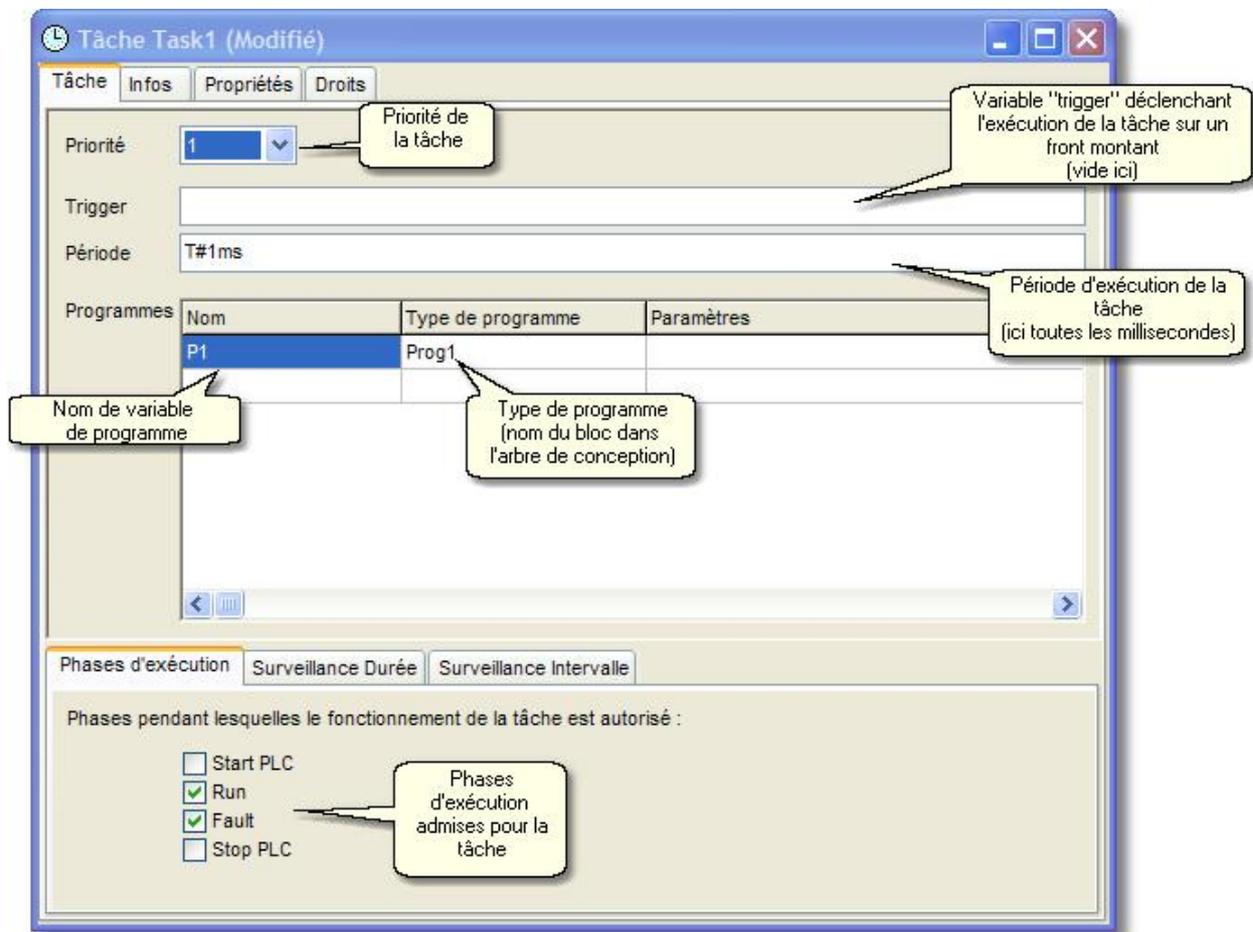
The proposed languages are the ST, the IL, the SFC (format text or diagram), the ladder, the blocks diagram and the Apigraf language.

The language is generally chosen at the creation of the POU, however it can be modified in this page. No conversion tool from a language to the other is available, the change of language implies the loss of all instructions of the POU therefore. A warning message is displayed at the language change, the change will only be efficient after the user's confirmation.

When a new language is chosen, the page "Instructions" is immediately selected.

2.3.1.5.5 Page Task

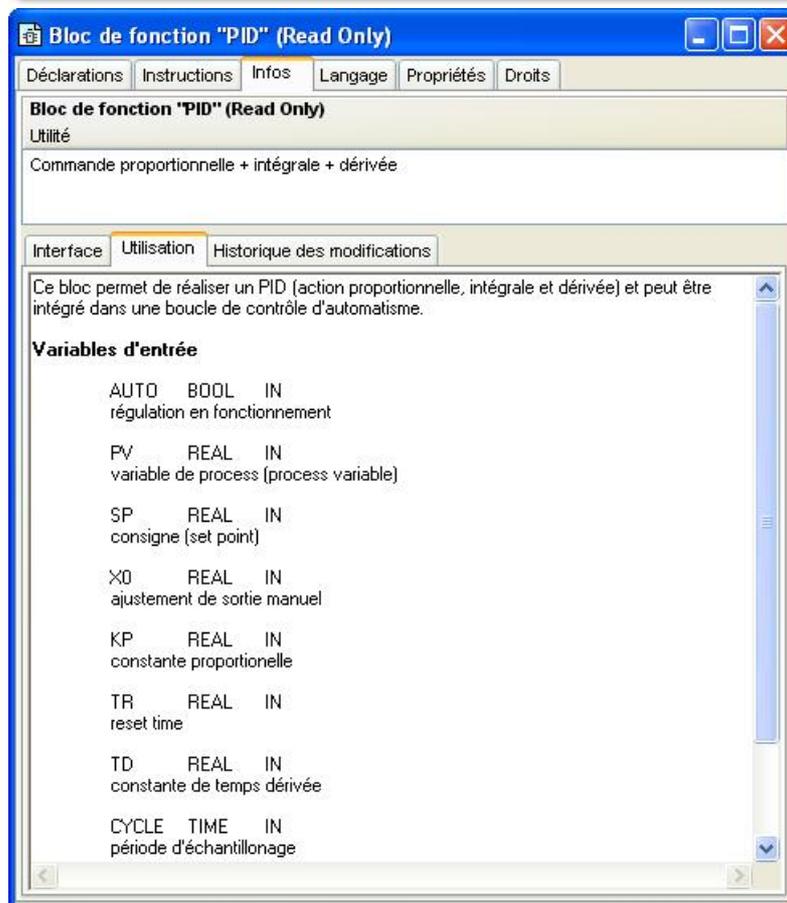
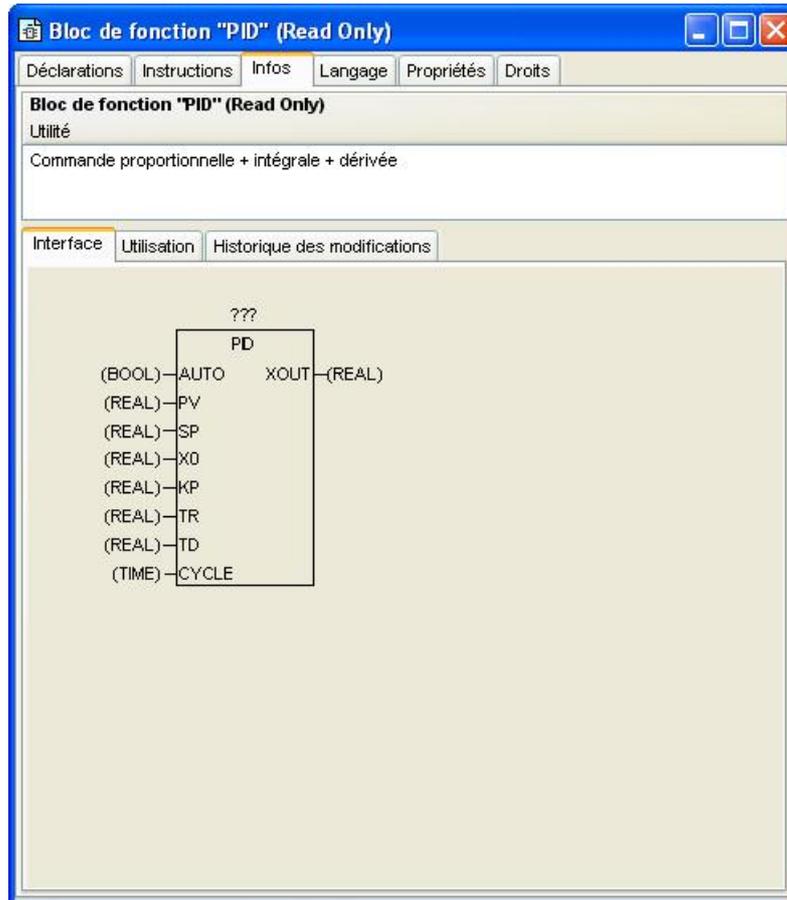
This page permits to affect the parameters associated to one [task](#)³².



- the priority level for the execution of the programs or function blocks
- the trigger
- the period (use the syntax of the norm IEC 61131-3 for time constants)
- the list of the programs or function blocks to run, presented as a table.

Besides, this page permits to specify the phases of execution admitted.

2.3.1.5.6 Page Infos



The first information includes the name and the type of theselected POU.

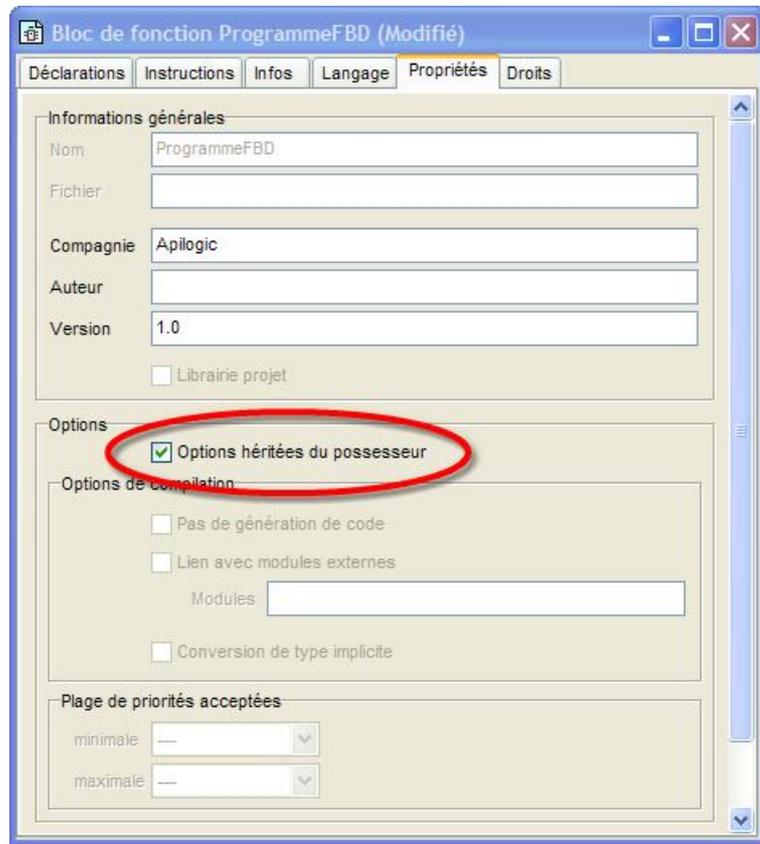
An edit zone permits to specify the general utility of it.

The tab page situated at the bottom of the page permits to display the general interface of the POU if it is about a function block or program or a function. The following tab permits to enter a descriptive of the use of the POU. Finally the last page is dedicated to the historic of the modifications.

2.3.1.5.7 Page Properties

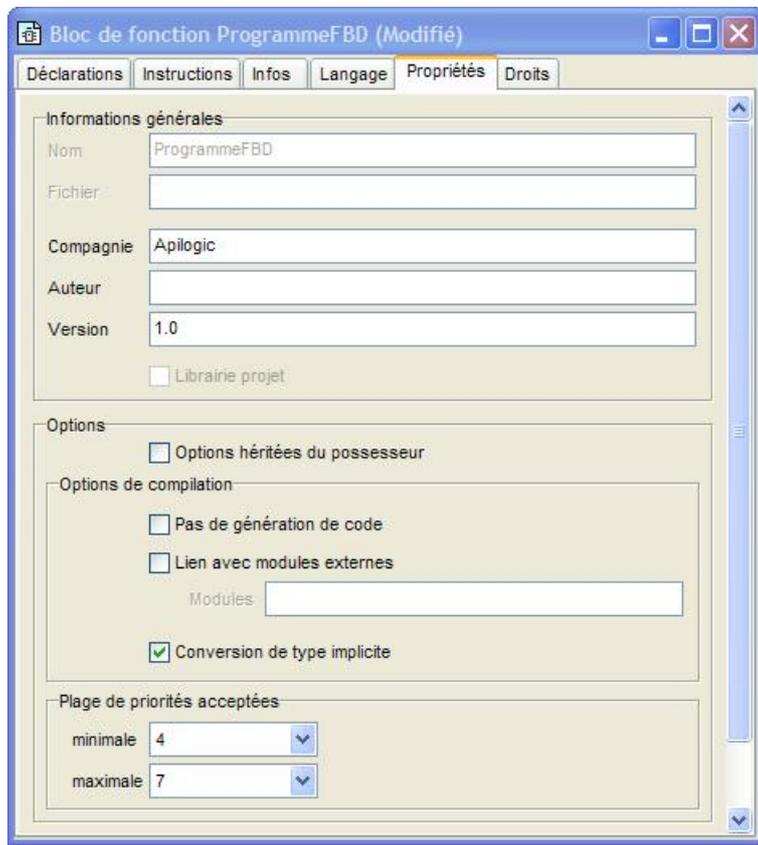
A part "general Information" permits to display the name of the POU and some traceability information concerning its conception (author, version).

In the case of a library, the name of the corresponding file is mentioned, and a checkbox permits to indicate if it is a ["project" library](#)³⁶ or no.



A part "Options" permits to define some compiling options and the priority levels accepted, by default these options are inherited from the owner, like is shown above on the copy of screen.

The fact to inherit the options permits to modify easily and simultaneously the options concerning the set of the POUs in a library or a subgroup. However it is possible to define the options POU by POU, by unchecking the box "inherited options".



In this case the following options are available :

- No code generation

This option permits to deactivate all code generation for this POU, that is therefore merely ignored by the workshop at the time of the compilation.

- Links with external modules

This option permits the use by a POU of the tools provided by an outside code library, non written in language IEC 61131-3. This possibility is especially exploited in the standard libraries of Optima PLC.

- Implicit type conversion

By default, the internal compiler of Optima PLC conforms to the rules voluntarily very strict of type control of the standard IEC 61131-3. It is not for example possible to affect an INT in a DINT without doing a conversion of the INT type explicitly toward DINT before affectation.

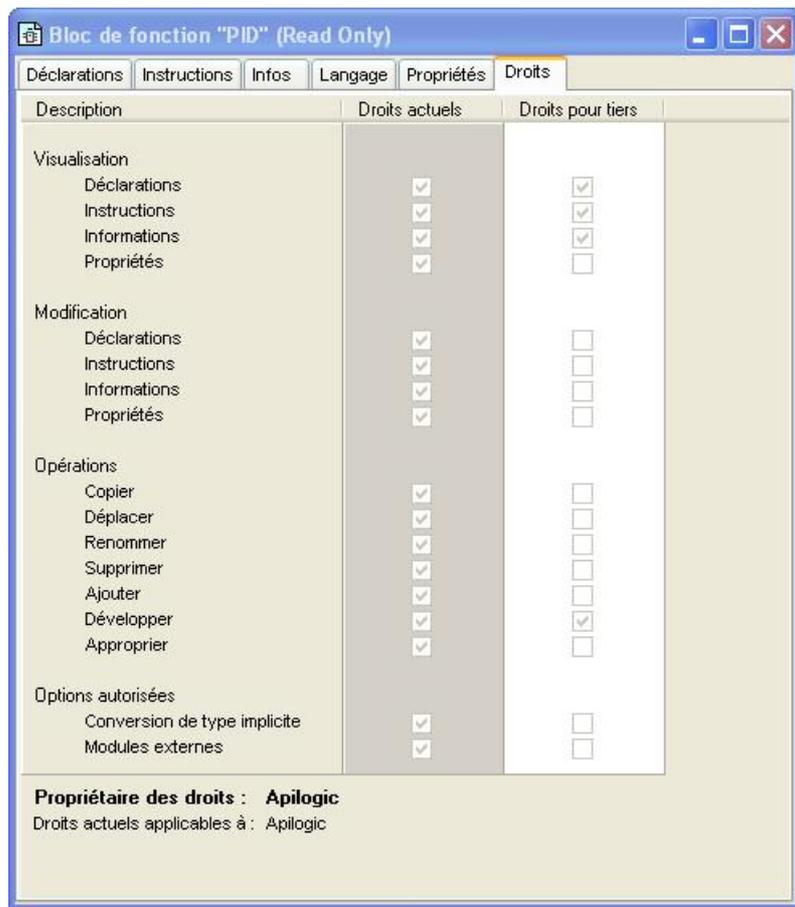
However, it is sometimes possible to lighten these constraints without particular risk, as in the above stated case; it is what this option permits, that, if it is checked, reduced the necessity to explicitly make type conversions.

- Range of allowed priorities

Some POU's cannot be used to all priority levels of the system. For example, it is easy to understand that a block achieving some accesses to a file on disk cannot be executed to the "real time" priorities.

The range of allowed priorities permits to define the priorities to which the system accepts the execution of the POU.

2.3.1.5.8 Page Rights



The access rights to the POU's are managed in this page.

The rights are classified according to 4 main criterias : visualization, modification, various operations and authorized options.

For the visualization and the modification, the rights come to apply to the properties, information, declarations and instructions.

The rights of operations concern the actions that apply on the entirety of the POU (addition, deletion.).

The rights on the authorized options come to limit the possibilities of compilation.

The page presents itself as a table that specifies what are the present rights (that is those of the present user) and the rights allowed for a third. It is possible in these conditions to keep the ownership of a POU by allowing another user to use it in his libraries while restricting him the possibilities of consultation or modification.

2.3.1.5.9 Table of the visible pages according to the POU's

Type/ Page	Infos	Properties	Rights	Task	Declarations	Instructions	Function Type
Task	X	X	X	X			
Global variables	X	X	X		X		
Types	X	X	X		X		
Block of program	X	X	X		X	X	
Function Block	X	X	X		X	X	
Function	X	X	X		X	X	X
subgroup	X	X	X				

2.3.1.6 Libraries Optima PLC

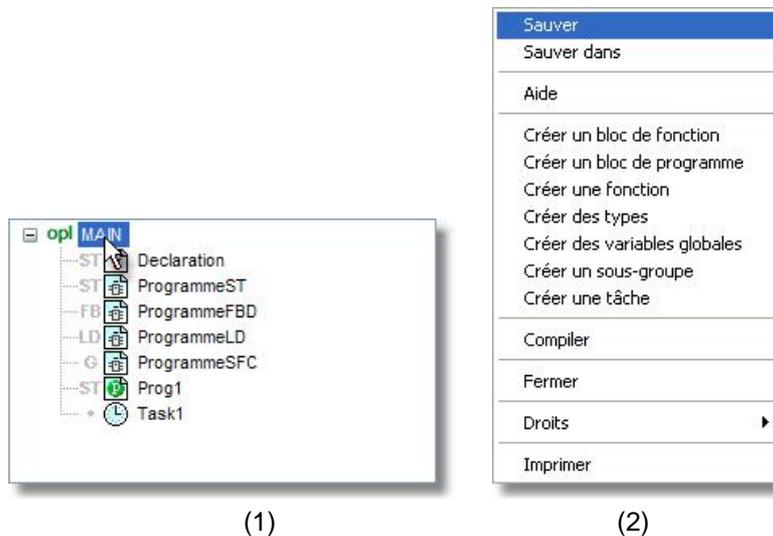
The software package Optima PLC contains standard libraries, only composed of functions or function blocks. Here is a preview of it, an exhaustive description of every library being available in this documentation.

SYSTEM	Tools systems. Give the possibility to leave or to generate an exception in the runtime and also to insert some information in the file of events of Optima PLC
STANDARD	Standard blocks and functions of the norm IEC 61131-3.
MESSAGES	Display of messages
IODRIVERS	Management of the drivers of inputs/outputs
TEXTOUT	Management of the output text files of Optima PLC
FILES	Management of the general files
SERIALCOM	Management of the serial ports
JACK	Piloting of actuator simple or double effect
TABSAMPLES	Use of sample tables
TCP	Communication in TCP

2.3.2 Creation of a library

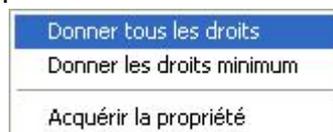
2.3.2.1 Menu of libraries management

To make this menu appear, select the library in tree (1) then click on the right mouse button (to make appear the popup menu associated to the libraries) (2).



The menu is composed of some parts :

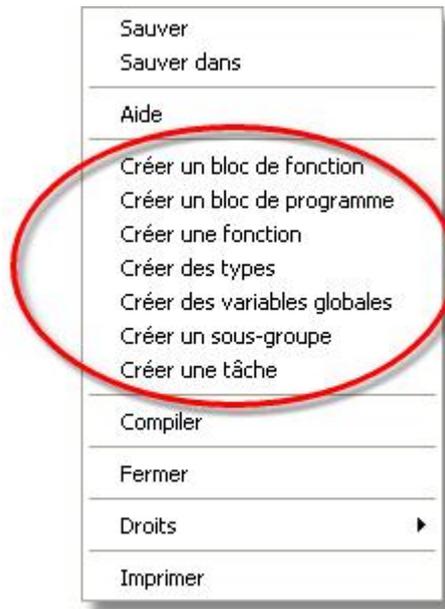
- To save or to rename a library (to Save and to Save in)
- To visualize the on line help
- To create a POU
- To compile the selected library
- To close the selected library
- To affect rights of maximum or minimum access at third on the POU and those that it contains (if you are the present rights owner of the POU).



- To print the programming documentation

2.3.2.2 Create a POU

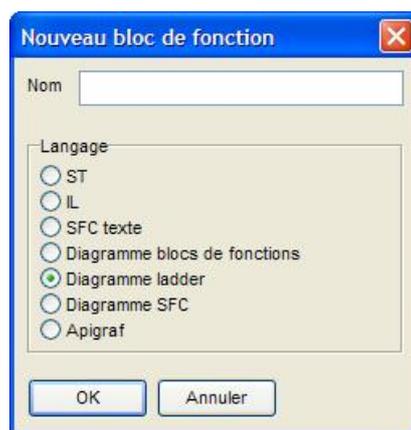
The menu associated to the libraries permits to create a POU among the 7 following types :



- Function block
- Block of program
- Function
- Types
- Global variables
- Subgroup
- Task

If the POU to create is the type "[Variable global](#)^[33]s", "[Types](#)^[33]" or "[subgroup](#)^[34]" then the icon associated to this type is immediately created in tree, the cursor is positioned on the name of the POU (by default, it is named "New").

If the POU is then of another type a window of dialogue appears to name the POU and to choose the programming language of this POU.



The icon associated to this new POU is created in tree only if a name has been given effectively to this POU and if the dialog box has been validated by the "OK" button.

2.3.2.3 Menu of POUs management

The popup menu associated to the selected POU is in any case function of the type of the POU, however the following options are available :

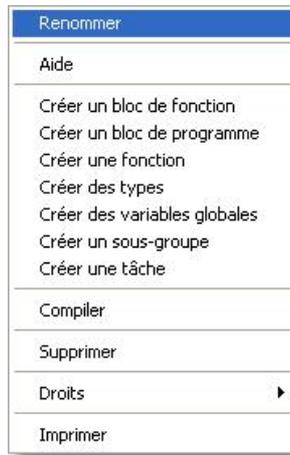
- Rename

- Display the contextual help
- Compile
- Delete
- Affect the rights
- Print the POU



Some supplementary options of menu are available when the selected POU is :

- A subgroup: it is then possible to create a new POU



- A function block: it is then possible to create an action ; in the present version this option is only available when the programming language is SFC diagram.



Remark : some options can be inhibited according to the user's rights.

2.3.2.4 Creation of Ladder diagrams

The low part of the tab page "Instructions" contains the diagrams of instructions. A column of headline indicates the number of diagram (or number of network). When a new diagram is created, it appears under the shape of a line however it is composed of 3 parts: an element of network beginning (selectable), an element of network end (selectable) and a line between these 2 elements.



To modify the equation of the network, the element of network beginning or a contact must be selected.



To modify the output, the element of network end or a coil must be selected.



Every coil or contact is named, this name corresponds to the variable associated to the element.

To select several elements at the same time, press the Control key of the keyboard at the same time as the selection done with the mouse (by simple click). It is also possible to select the elements contained in a zone of the display by stretching a rectangle including selection (click a corner of the rectangle while leaving the mouse button pushed, displace the mouse to the opposite corner, and release the mouse button).

The available toolbar in top of the page "Instructions" permits to add some elements or to modify the features of the element selected. A popup menu is associated to every selected element, the options that it proposes are different according to the type of element.

2.3.2.4.1 Toolbar of the ladder diagrams

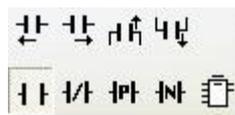
The toolbar consists of 6 parts :



- Management of contacts
- Management of the outputs
- Negation
- Management of the diagrams
- Various operations
- Zoom and printing (in a WMF file or toward a printer)

2.3.2.4.1.1 Management of contacts

Nine buttons permit to add / modify contacts :



The five lower buttons can be used of two different manners: if a contact is selected, clicking one of these buttons transforms contact according to the icon of the button ; and if a drag-and-drop operation is initiated from the button (click, maintain the mouse button pushed, then release on already existing contact), a new contact corresponding to the icon is inserted.

	permits to insert a contact before contact or contacts selected (or in beginning of line if no contact is present in the diagram)
	permits to insert a contact after contact or contacts selected (or in beginning of line if no contact is present in the diagram)
	permits to insert a contact in parallel and above contact or contacts selected
	permits to insert a contact in parallel and below contact or contacts selected
	<ul style="list-style-type: none"> • permits to transform the type of contact selected in negative contact. • a drag-and-drop with this button permits to insert a

	negative contact on the left of the element on which it is released.
	<ul style="list-style-type: none"> permits to transform the type of contact selected in normal contact. a drag-and-drop with this button permits to insert a normal contact on the left of the element on which it is released.
	<ul style="list-style-type: none"> permits to transform the type of contact selected in active contact on rising edge. a drag-and-drop with this button permits to insert a contact active on rising edged on the left of the element on which it is released.
	<ul style="list-style-type: none"> permits to transform the type of contact selected in active contact on falling edge. a drag-and-drop with this button permits to insert a contact active on falling edge on the left of the element on which it is released.
	<ul style="list-style-type: none"> permits to transform contact selected in function block, by default adjusting makes itself on the first of the inputs and the first of the outputs of the block. a drag-and-drop with this button permits to insert a default block "&" on the left of the element on which it is released.

2.3.2.4.1.2 Management of the coils

The following buttons permit to add / modify the outputs :



The lower buttons can be used of two different manners: if a coil is selected, clicking one of these buttons transforms the coil according to the icon of the button; and if a drag-and-drop operation is initiated from the button (click, maintain the mouse button pushed, then release it on a already existing coil), a new coil corresponding to the icon is inserted.

	permits to insert a coil in parallel and above the selected one(s)
	permits to insert a coil in parallel and below the selected one(s)
	<ul style="list-style-type: none"> permits to insert a coil, for which the state of the associated variable is the result of the equation of contacts that precedes it. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.
	<ul style="list-style-type: none"> permits to insert a coil, for which the state of the associated variable is the contrary of the result of the equation of contacts that precedes it. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.
	<ul style="list-style-type: none"> permits to insert a coil, for wich the state of the associated variable becomes "true" when there is a rising edge on the result of the equation of contacts that precedes it. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.
	<ul style="list-style-type: none"> permits to insert a coil, for which the state of the associated variable becomes "true" when there is a falling edge on the result of the equation of contacts that precedes it. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.

	<ul style="list-style-type: none"> permits to insert a coil, for which the state of the associated variable becomes "true" when the result of the equation of contacts that precedes is true. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.
	<ul style="list-style-type: none"> permits to insert a coil, for which the state of the associated variable becomes "false" when the result of the equation of contacts that precedes is true. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.
	<ul style="list-style-type: none"> permits to transform the selected coil in function block. A drag-and-drop with this button on a coil of the diagram inserts this type of coil rightly before.
	<ul style="list-style-type: none"> permits to transform the selected coil in "return" block (come out of the block). A drag-and-drop with this button on a coil of the diagram inserts a return just before.
	<ul style="list-style-type: none"> permits to transform the coil selected in jump block ("goto") toward a label defined in the present block. A drag-and-drop with this button on a coil of the diagram inserts a just jump before.

2.3.2.4.1.3 Negation

When the input or the output of a function block is selected, it is possible to affect it a negation with the button



2.3.2.4.1.4 Management of the diagrams

It is possible to define several diagrams in a POU :

-  permits to add a diagram before the one that is selected
-  permits to add a diagram after the one that is selected

2.3.2.4.1.5 Various operations

The buttons placed at the end of the toolbar permit :

-  permits to make a zoom on the selected function block, ie to visualize its instructions when user's rights permit it.
-  permits to cancel the last modification done in the diagrams.
-  : these three icons permit to cut, to copy and to paste the selection respectively.

2.3.2.4.1.6 Zoom and printing

It is possible to make a zoom in or out to better visualize the diagrams :

-  : shorten the display of the diagram
-  : enlarges the display of the diagram

The instructions can be saved in a file or printed :

-  : save in a WMF file
-  : printing of the diagrams

2.3.2.4.2 Use of the selection

The insertion of contact in parallel with other contacts (function "or") requires that these other contacts are selected before.

Example: to achieve a diagram testing the condition (L1 & L2) or L3. Initially L1 contacts and L2 have been inserted like follows.



Select L1 and L2 by clicking them while pressing the Control key



Click on the button  : a new contact is inserted in parallel with the selection. It remains to hit the name of the associated variable.



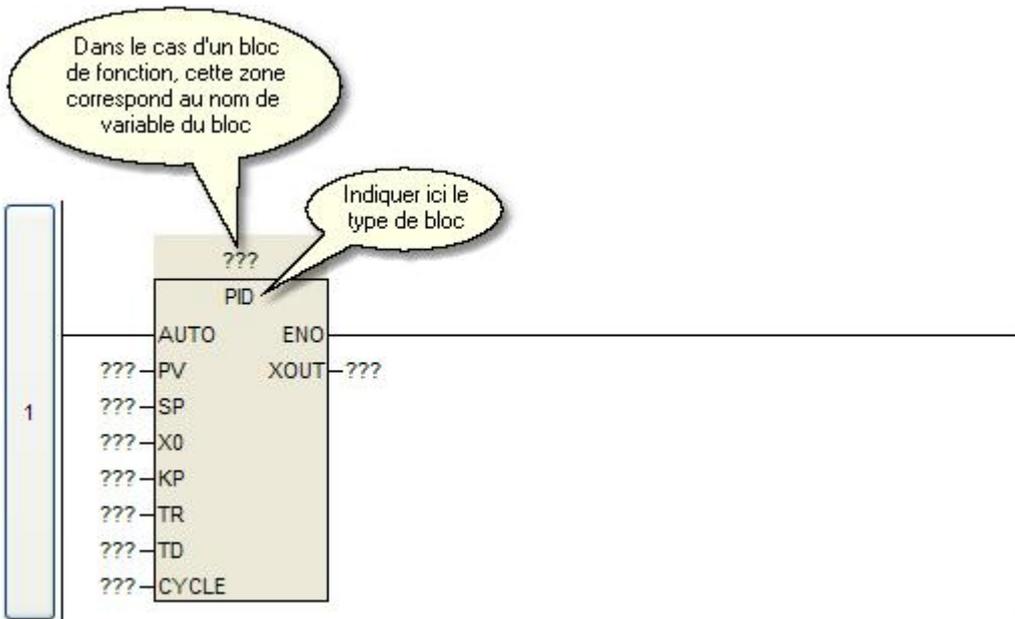
2.3.2.4.3 Use of blocks in a ladder diagram

Some blocks (functions and function blocks) can be used in a ladder diagram, while respecting some rules.

1st method: use icons

To insert a block in contact, insert a normal contact first of all, with for example the button , then transform this contact in block by clicking on the button  (it is as possible to directly drag the button  on the position of insertion of the block). Finally, enter the type of block wished.

In the following example, a default block & is inserted, then replaced by a PID (this type is hit instead &).



In the case of a function, only the type of block must be written. If it is a function block, a supplementary input box situated over the block permits to enter the name of the correspondent variable, that must be declared with this type of block.

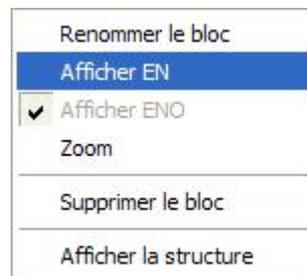
Since the version 2.9 of Optima PLC, the declaration of the variable corresponding to a function block is no more obligatory. This one is automatically made with a default name.

2nd method: use tree of POU

Click on the tab Libraries, and develop the part of the tree from which you wish to insert a block. Click on the block while maintaining the mouse button pushed, then release it at the wished insertion site. In this manner, a PID block can be inserted for example directly from the Standard library.

Use of EN and ENO (Enable, EnableOut)

At the insertion of a block, the first input and the first output are normally connected to the ladder network. However, these inputs and outputs are not necessarily of BOOL type, as well as the ladder network would like it. In this case, a supplementary input EN and/or a supplementary output ENO are added. Use of EN or ENO can also be forced (contextual menu of the block) :

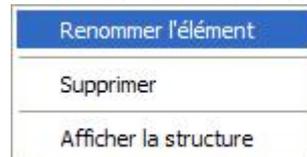


Above, in the illustrated case the PID block includes a first input Auto of BOOL type, that is connected to the network ladder. On the other hand its first XOut output is of type REAL: an ENO output is automatically added.

2.3.2.4.4 Contextual menus of the ladder diagrams

The menus are function of the element selected that can be :

- a contact or a coil: the selected element can be renamed or suppressed



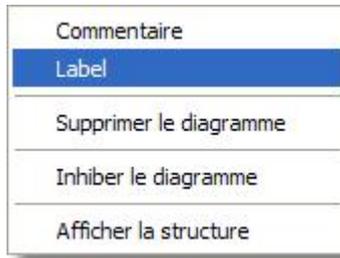
- a text: the options of standard edition are proposed



- a block, an input (connected or no) or an output of a block :



- a diagram: it is possible to add a label or commentary to a diagram, or to suppress the selected diagram



2.3.2.4.5 Example of creation of a ladder diagram

The equation to represent is the following : PowerOn = (Start1 or Start2) and not EmergencyStop

Select (or create) a POU of Function block type in ladder language.

The first stage to do is to create the variable of the equation.
 Select the tab "Declarations" and insert the following declarations :

```

Var_Input
  Start1      : bool;
  Start2      : bool;
  EmergencyStop : bool;
End_Var

Var_Ouput
  PowerON     : bool;
End_Var
    
```

Select tab "Instructions".

Select the input element of the diagram, click on and name the contact "Start1".

Select contact "Start1", click on and name the new contact "EmergencyStop".

Select contact "EmergencyStop" then click on .

Select contact "Start1", click on and name the new contact "Start2".

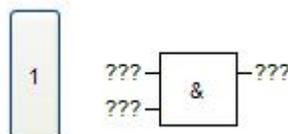
Select the output element of the diagram, click on and name the new coil "PowerON".



The diagram is complete. To verify that no error has been committed during the creation, compile it.

2.3.2.5 Creation of FBD diagrams

The low part of the tab page "Instructions" contains the instruction diagrams. A column of headline indicates the number of diagram. When a new diagram is created, it is composed of only one block (of type AND).



It is from then possible to modify this first block or to add some blocks upstream and downstream in the diagram (connected to the inputs and outputs of the block).

To select several blocks at the same time, press the Control key of the keyboard and at the same time select blocks with the mouse (by simple click).

The available toolbar in top of the page "Instructions" permits to add some diagrams, to apply a negation on an input of a block and to visualize the internal instructions of the selected block.

A popup menu is associated to every selected element, the options that it proposes are different according to the type of element.

2.3.2.5.1 Toolbar of a FBD diagram

The buttons of the toolbar are active according to the selected element :

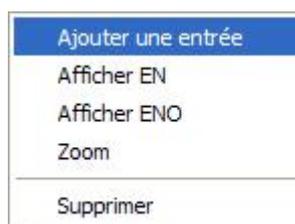


	when a diagram or one of its elements (block, input or output of a block) is selected, it is possible to add a diagram before the diagram concerned by the selection
	when a diagram or one its elements (block, input or output of a block) is selected, it is possible to add a diagram after the diagram concerned by the selection
	when a block is selected, it is possible to visualize the internal instructions of it
	when an input or an output of block is selected, it is possible to apply a negation
	this button permits to cancel the last modification done in the diagram
	copy the selection in the clipboard and suppresses it of the diagram
	copy the selection in the clipboard
	paste the clipboard before the present selection
	shorten the display of the diagram
	enlarges the display of the diagram
	save the diagram in a WMF file
	printing of the diagram

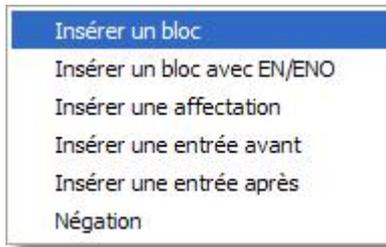
2.3.2.5.2 Contextual menus of a FBD diagram

The menus are function of the selected element that can be :

- a block: display the input/output of validation of the block (EN and ENO in the norm IEC 61131-3), make a zoom on the internal instructions of the block, suppress the block, add it an input if the number of inputs of the block is variable (example: one AND can be done on 2 inputs, 3 or more), rename the block when it has a name (example: PID).



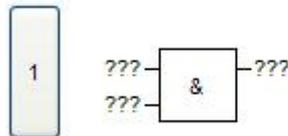
- an input of block : add a block (with or without the input/output EN and ENO), insert a branch of derivation (affectation), apply to it a negation or, in the case where the number of inputs is variable, add an input or to suppress the one that is selected.



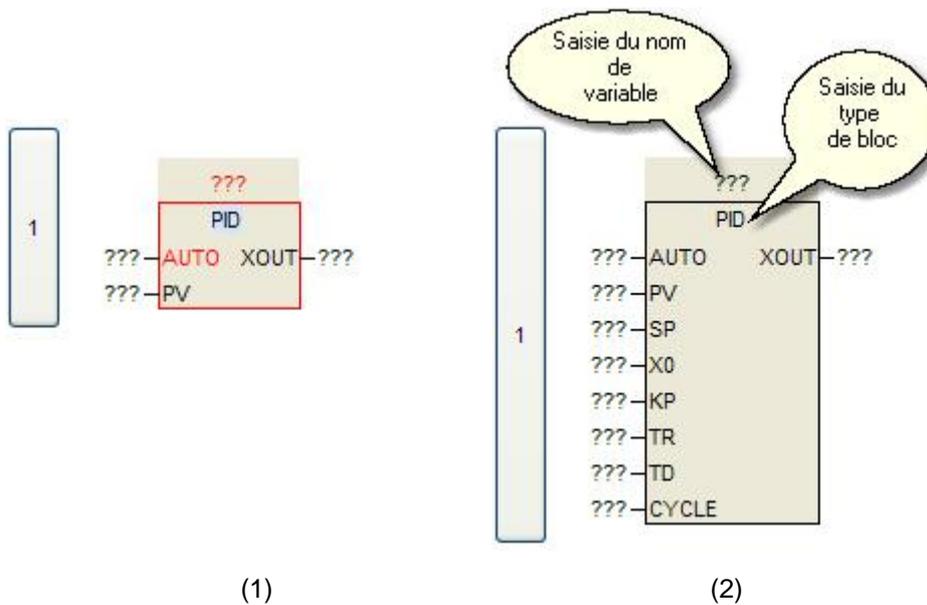
- an output of block or an output of a derivation branch : add a block (with or without the input/output EN and ENO), insert a branch of derivation (assignment) or to apply to it a negation.
- a branch of derivation (assignment): to add an output on this derivation
- a diagram: it is possible to add a commentary or a label to a diagram, or to suppress the selected diagram.
- a text: the options of standard edition are proposed, it is to note that modifying the text corresponding to the function of the block entails a complete update of the block.

2.3.2.5.3 Definition of the type of block

At the insertion, the block takes the default type "&" correspondent to the AND function.



To modify the default type, edit the text corresponding to the type of block (here &) by clicking over, then by hitting the new wished type.



- (1): the name of PID block is hit, the block gets partially up to date during the striking.
- (2): The name is validated, the block gets up to date (addition or deletion of inputs / outputs according to its interface).

If the block is a function block or program, it is necessary to enter the name of the correspondent variable in the input zone that is rightly above. The correspondent variable must be defined in the declaration part, or in a declaration of global variables. Since the version 2.9, this declaration is no more obligatory : leaving empty this input zone, the declaration is automatically done with a default name.

If the block is a function, this input box situated over the block doesn't exist.

2.3.2.5.4 Use of variables at input and output of block

A variable can be used for every input or output of block. This variable must be the same type that the input or the output.

For the inputs, it is also possible to use a ST expression whose result corresponds to the type of the input.

To define a variable, click on the input box and enter the wished name.

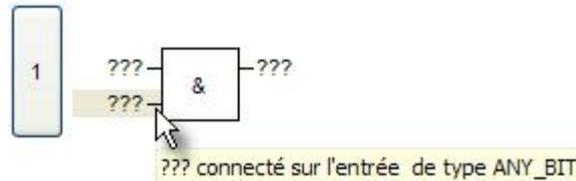
2.3.2.5.5 Chaining blocks

The blocks can be chained while respecting the following rules :

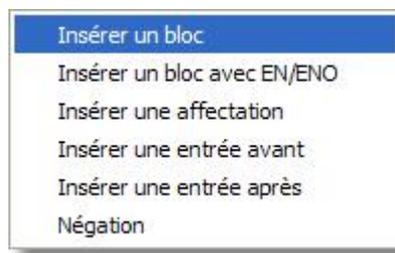
- only the first output of a block can be chained to the input of another block.
- all inputs of a block can be chained to the first output of another block.

To chain a new block on an input or the first output of a block, proceed as follows :

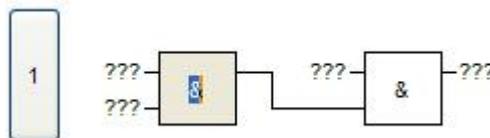
Make a right click on the input or output concerned.



The contextual menu is displayed :



Click on "insert a block", the default block is then added :



This block can be modified by hitting the type of block wanted.

2.3.2.5.6 Example of FBD diagrams

The equations to represent are the following :

DemandStart = Start1 or Start2
PowerON = DemandStart and not EmergencyStop

Select (or create) a POU of type Function block in FBD diagram language.

The first stage to do is to create variable of the equation.

Select the tab "Declarations" and insert the following declarations :

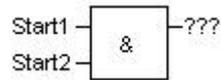
```
Var_Input
  Start1      : bool;
  Start2      : bool;
  EmergencyStop : bool;
End_Var
```

```
Var_Output
  PowerON      : bool;
  DemandStart  : bool;
End_Var
```

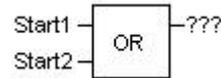
Select the tab "Instructions."

Select the block "&" created at the same time as the diagram, click on one of the inputs of the block and, by using the contextual menu, affect the variable "Start1" (by modifying the text "???" of the input).

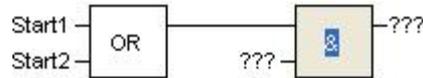
Select the second input and affect the variable "Start2."



Select the block "&" and modify its function by clicking on the text & and by replacing it by "OR".

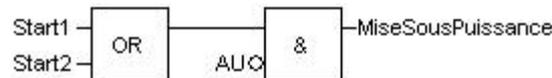


Select the output of the block and insert a block.



Select the input of this second block that is not joined to the precedent and apply to it a negation (by using the contextual menu) then the variable EmergencyStop.

Select the output of the second block and affect it the variable "PowerON".

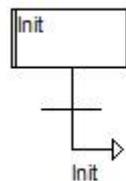


Select the output of the first block and with the help of the contextual menu "Affectation", add a parallel output to affect the variable "DemandStart."

The diagram is complete. To verify that no error has been committed during the creation, compile it.

2.3.2.6 Creation of SFC (Grafcet) diagrams

The low part of the tab page "Instructions" contains an instruction diagram (and one only, to create another SFC diagram, it is necessary to create another POU).



When a new diagram is created, it is composed of an initial step (named Init by default), a transition and a branch that sends back toward the initial step.

The available toolbar in top of the page "Instructions" permits to add some elements in the diagram.

The deletion of elements must be logical and must leave a coherent diagram. Thus, if a step is selected alone, it cannot be suppressed, the deletion is only possible if, for every selected step, there is at least a selected transition at the same time.

The multiple selection is made by pressing the Control key of the keyboard as the element is selected with the mouse. At the insertion of a divergence "AND" or "OR", the divergence is put in parallel with the set of the selected steps.

Transitions

The conditions of transition must be defined as a condition according to the ST syntax. The input is made by clicking the transition zone on the right.

A transition can be left empty, in this case it is considered like being always true.

Some commentaries can be placed in a transition, with the classic syntax: //, {} or (* *). Attention however, if the transition only contains a commentary, it is not considered like being by default true, and an error of compilation is signalled. To avoid it, to put the True mention outside of the commentaries.

Stages

The edition of the step name is made by clicking the content of the box (in top on the left). By default, a new step doesn't include a name, this one is not indeed obligatory, except for the jumps.

A step can have one or several actions, however the addition of an action is not obligatory, a step without action is said "empty step". The display of the action can be hidden/visible thanks to the buttons  and  situated in the rectangle of step.

The definition of an action can be made several ways :

- Boolean action: a boolean variable is specified, the action consists then to set to 1 this variable.
- Action "in situ" : the action is programmed directly in the diagram, it must be defined in one of textual languages.
- External action: the action mentioned in the diagram makes reference to an action defined outside of the diagram, under the dependence of the current block, and therefore must be at the lower level in the conception tree. That actions can be defined in any language.

A popup menu is associated to every selected element, the options that it proposes are different according to the type of element.

2.3.2.6.1 Toolbar of a SFC diagram

The buttons of the toolbar are active according to the selected element :



	when a step, a transition or a back branch is selected, it is possible to add a step/transition before the selected element
	when a step or a transition is selected, it is possible to add a step/transition after the selected element
	when a transition is selected, it is possible to add a parallel branch (function OR) on the left of the selected transition
	when a transition is selected, it is possible to add a parallel branch (function OR) on the right of the selected transition
	when a step is selected, it is possible to add a parallel branch (function AND) on the left of the selected step
	when a step is selected, it is possible to add a parallel branch (function AND) on the right of the selected step
	when a transition belonging to a parallel branch is selected, it is possible to modify the return, that is to redirect it toward another step
	hide all actions of all steps
	unmask all actions of all steps
	enlarges the display of the diagram
	shorten the display of the diagram
	print the diagram
	export the diagram in the directory of the project as a WMF file whose (Windows Meta File) name is the name of the block.
	this button permits to cancel the last modification done in the diagram

2.3.2.6.2 Contextual menus of the SFC diagrams

The menus are function of the element selected that can be :

- a step: the step can be renamed or an action can be added

Renommer l'étape
Ajouter une action

- a transition: the condition of passage of the transition can be entered "in situ"

Editer la transition in situ (ST)

- a branch back: to rename the step toward which the branch returns
- an action: to parameterize or to suppress the selected action
- a text: the options of standard edition are proposed

2.3.2.6.3 Creation of an action

The creation of an action is made in 2 stages :

- parameterization of the action
- implementing of the action

An action is represented by a set of rectangles placed on the right of the step, the first to display the qualifier, the second (to its right) for the name of the action, and possibly a third below the 2 first for the instructions definite in situ.

To create an action, make a right click on the step ; the contextual menu is displayed :

Renommer l'étape
Ajouter une action

Click on Add an action. The following window is displayed :

Paramètres d'action

Qualificateur: N (dropdown) Temps: []

Variable d'action booléenne: []

Nom d'action: [] (dropdown)

Variable indicatrice: []

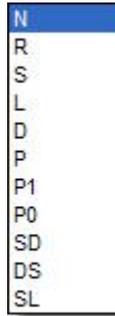
Action éditée in situ:

- En langage ST
- En langage IL
- En langage Apigraf
- Appel d'action inhibé
- Simple commentaire

Buttons: OK, Annuler, Aide

2.3.2.6.3.1 Qualifier of action

The field "qualifier" permits to define how the action will be executed. It is valid for all types of definition of action.



The different qualifiers are defined by the norm IEC 61131-3. Optima PLC gives you the possibility to use all.

Attention: the use of some qualifiers, in particular those permitting to temporize the actions, can complicate the working of the SFC diagram and make the debug difficult. The N qualifiers, R, S, P1 and P0 are generally sufficient.

Qualifier	Use	Parameter
N	Non memorized	
R	Reset to 0	
S	Set (memorized)	
L	Limited in the time	time
D	Deferred	time
P	Pulse	
P1	Pulse at the input in the step	
P0	Pulse at the output of the step	
SD	Memorized and deferred	time
DS	Deferred and memorized	time
SL	Memorized and limited in the time	time

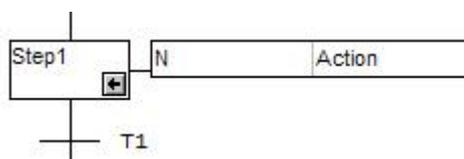
Attention, Optima PLC respects strictly the definition of the qualifiers of the norm IEC 61131-3. It implies that the actions are executed two times systematically to the minimum : as long as the condition of execution is true, and once again when it becomes false. The goal is to allow the action to make a particular treatment at the time of its deactivation. The action has a variable X indicating its activity. This variable changes to false when the action is deactivated.

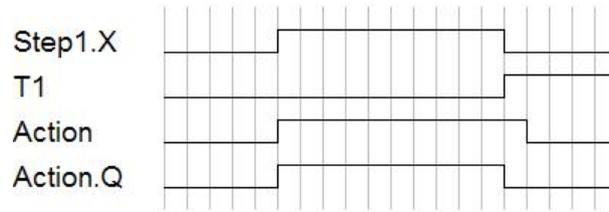
It especially poses problem with the P (impulse) qualifier. The action so qualified is executed at the entry in the step, but also a second time. The first time with the variable of X activity to true, the second time to false. A frequent error is to consider that the action will only be executed once, and to use this qualifier to do a numbering, for example, without testing the variable X to verify that the action is well active.

To palliate this problem, the norm introduced two exceptions : the qualifiers P0 and P1. The actions using these qualifiers are executed only one time, and are therefore the ideal way to treat an impulse at the input or at the output in a step.

N: Action non memorized.

This type of action call is the simplest: the action is executed as long as the step of call is active, then a last time after its deactivation





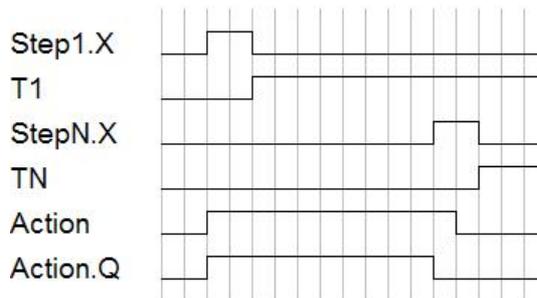
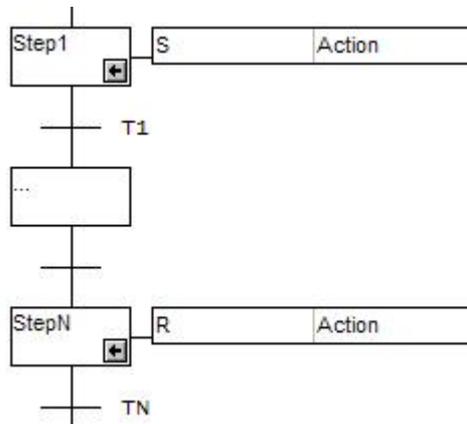
S: Action memorized (Set).
R: Action reset to 0 (Reset).

An action called with the S qualifier is executed as soon as the step becomes active, and remain memorized. It is executed until an ulterior call with the R qualifier, even though the initial call step is no more active.

If an action called with the S qualifier is never stopped with a R call, it is executed indefinitely at every execution of the block containing the SFC diagram.

When the action is called with the R qualifier, it is executed once again with the variable Q to 0.

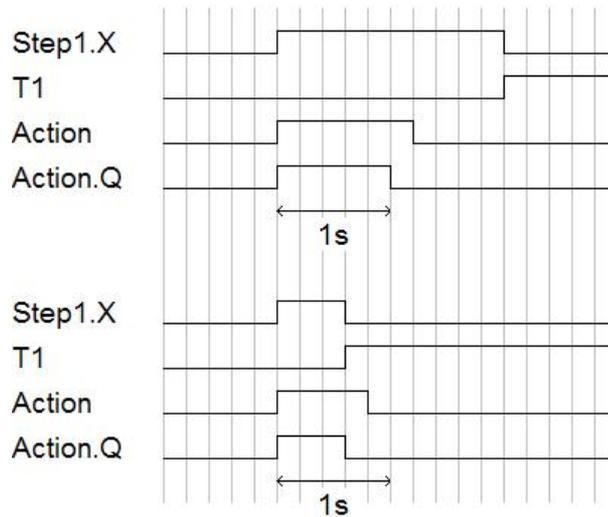
The R qualifier permits to stop an action called with another qualifier that S and memorizing an action (SD, DS or SL).



L: Action Limited in the time

An action called with the L qualifier begins its execution as soon as the step becomes active. It continues to execute during a specified lapse of time. If the step becomes inactive before the end of the delay, the action is stopped.

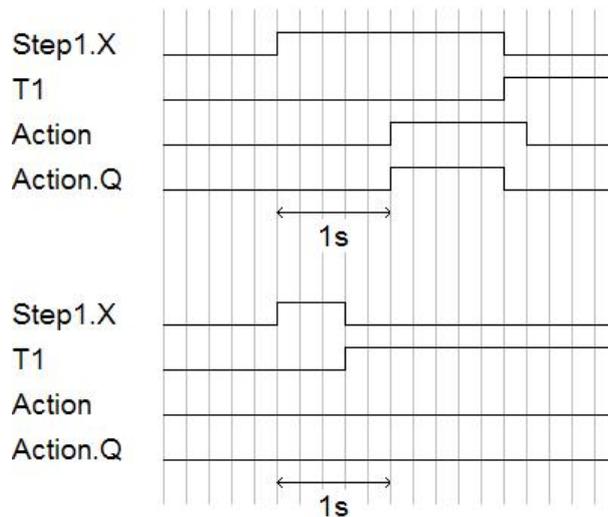




D: Deferred action

An action called with the D qualifier begins to execute after the end of a specified delay. It continues as long as the step remains active, then a last time after the deactivation of the step.

If the step becomes inactive before the end of the delay, the action is not executed.

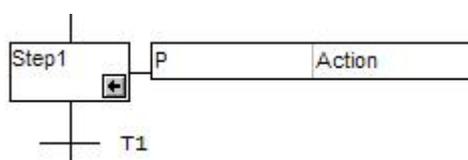


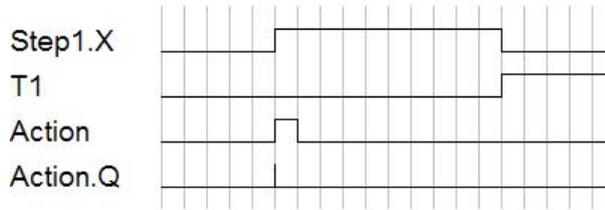
P: Action "Pulse"

The action called with the P qualifier is executed a first time at the activation of the step, then one second, this time with the variable Q to 0.

This type of call of action must be used with precaution: a frequent error consists in using it while thinking about wrong that the execution is made once, for a numbering for example : it is necessary in this case to use P1.

To do an impulse on a boolean variable (boolean action), the call of P action is the ideal: at the time of the first execution, the variable is put to 1, then handed to zero at the time of the second execution.



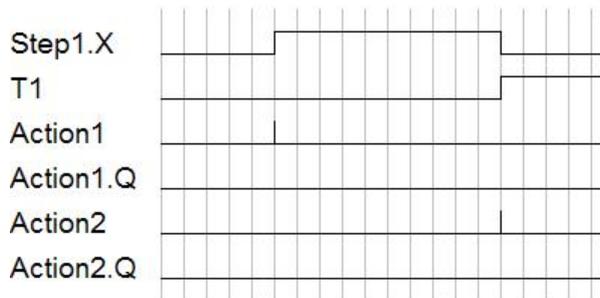
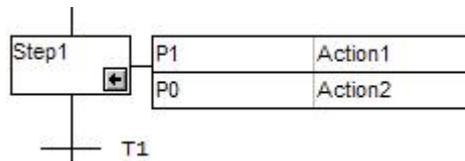


P1: pulse at the input in the step
P0: pulse at the output of the step

The actions called with these qualifiers are called only once, respectively at the activation and at the deactivation of the step.

The call being made once, the variable Q remains to 0. It also means that these qualifiers don't function with the boolean actions (the boolean variable recopies the state of Q).

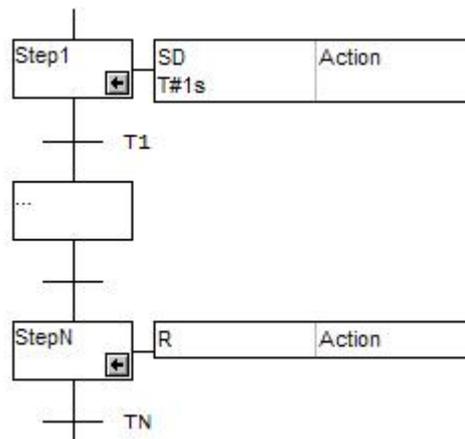
These calls of action are especially adapted to the actions including the numbering for example, or all treatment to execute only one time. P1 constitutes a good replacement for P.

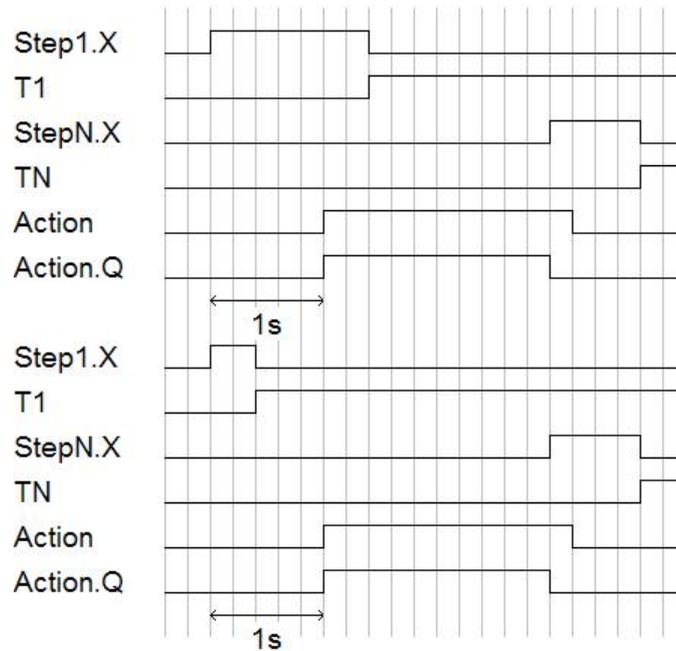


SD: Action memorized and Deferred

With this type of call, an action is memorized and begin to execute after a specified delay, even though the step is no more active.

The execution ends at an ulterior call of the action with the R (Reset) qualifier. If the call with R is done before the end of the delay, the action is not executed.



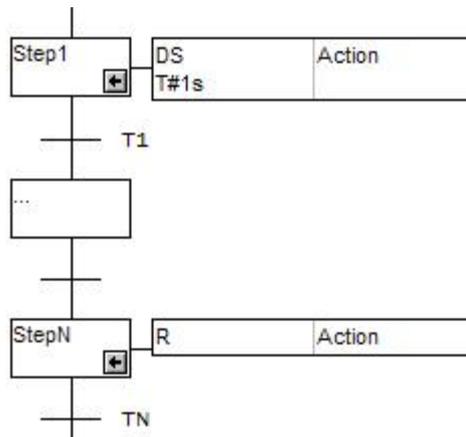


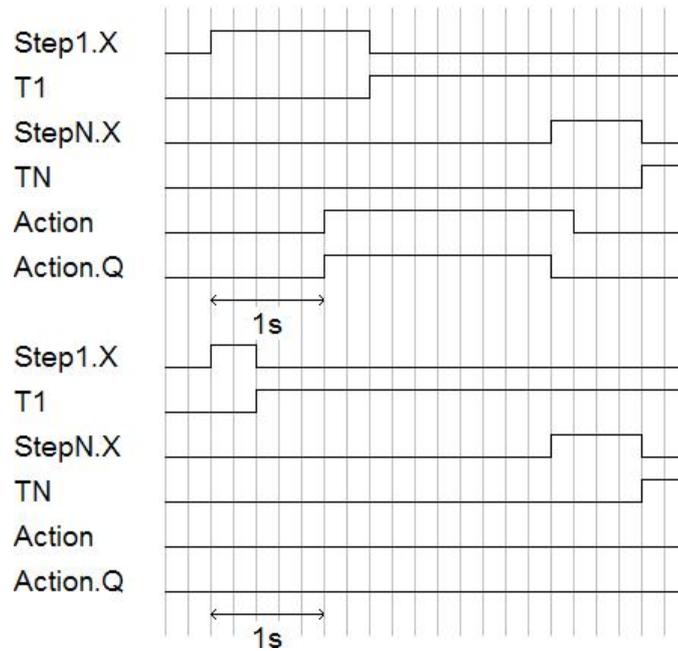
DS: Deferred and memorized action

The action called with the DS qualifier is deferred according to a specified delay. At the end of this delay, the action begins to execute, and is memorized : the execution continues even though the step is deactivated.

The execution ends at the time of an ulterior call of the action with the R (Reset) qualifier.

If the step is deactivated before the end of the delay, the action is not executed and is not memorized.

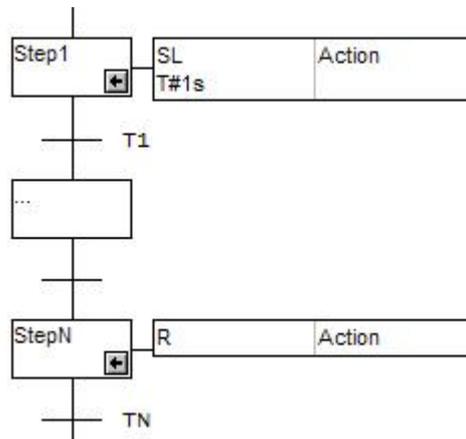


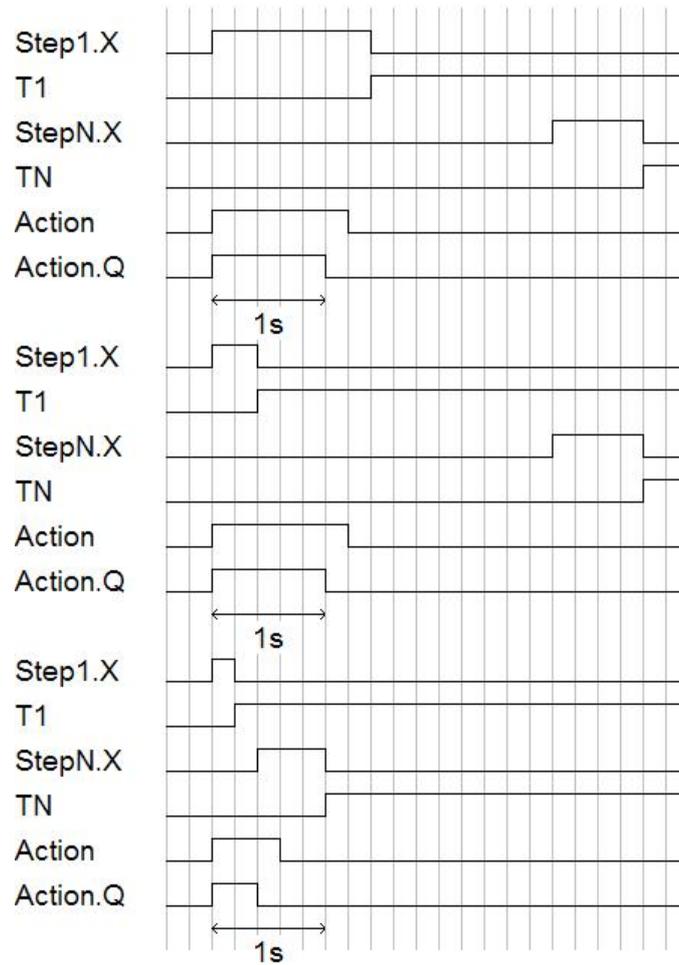


SL: Action memorized and Limited in the time

With this qualifier, an action is immediately memorized and to execute at the activation of the step. The action executes during the indicated delay, even though the step is deactivated.

The execution also ends at the time of an ulterior call of the action with the R (Reset) qualifier.





2.3.2.6.3.2 Parameters of a boolean action

In the case of a boolean action, the name of the concerned variable must be entered. The qualifier, although it can take all possible values, will generally be one of these :

N	The variable is set to 1 during the activation of the step, reset to 0 then as soon as the step is no more active.
S	The variable is set to 1 since the input in the step.
R	The variable is set to 0 since the input in the step.

If for example the variable SA1 must be set to 1 during the activation of the step :

Paramètres d'action

Qualificateur: N

Temps: []

Variable d'action booléenne: SA1

Nom d'action: []

Variable indicatrice: []

Action éditée in situ:

- En langage ST
- En langage Ladder
- En langage Apigraf

Appel d'action inhibé

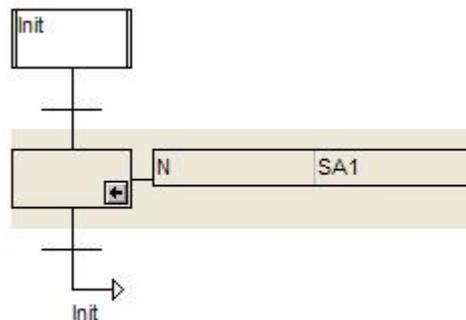
Simple commentaire

OK Annuler Aide

Champ de saisie de la variable booléenne

Ces champs sont inhibés si une variable booléenne est précisée

Here is the result :



2.3.2.6.3.3 Parameters of an action "in situ"

To define an action "in situ", that is to say written directly in the diagram, check the box corresponding to the wished textual language :

Paramètres d'action

Qualificateur: N

Temps: []

Variable d'action booléenne: []

Nom d'action: Test

Variable indicatrice: []

Action éditée in situ:

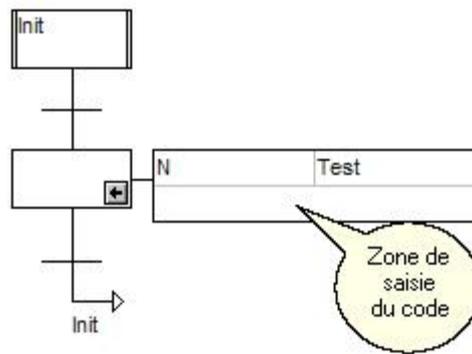
- En langage ST
- En langage Ladder
- En langage Apigraf

Appel d'action inhibé

Simple commentaire

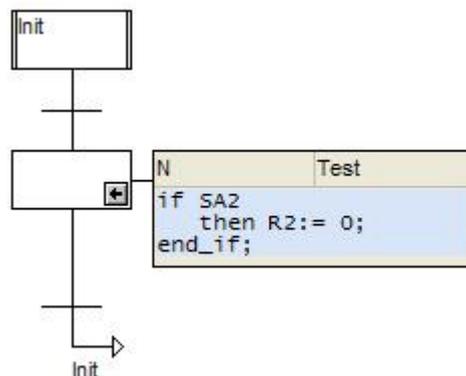
OK Annuler Aide

The field "name of action" can be left empty, the action won't be named in this case. However, an action, even defined "in situ", can be called in several places of the diagram. In this case, it must be named.



When the window is validated by OK, the empty action is inserted in the diagram. To enter the code, click on the input box of the code. The code entered here must be in ST language, because it is the language checked higher.

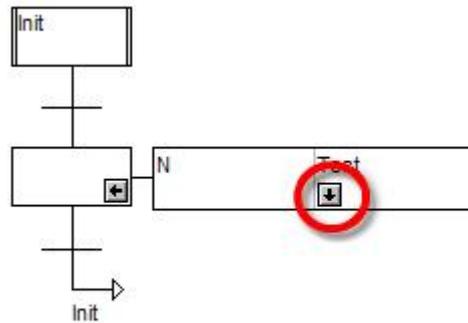
Example :



2.3.2.6.3.4 Parameters of a "external action"

To call a "external" action (that is to say non definite in the diagram directly), enter the wanted name in the field "name of action" only.

The call of the action is added to the diagram as follows.

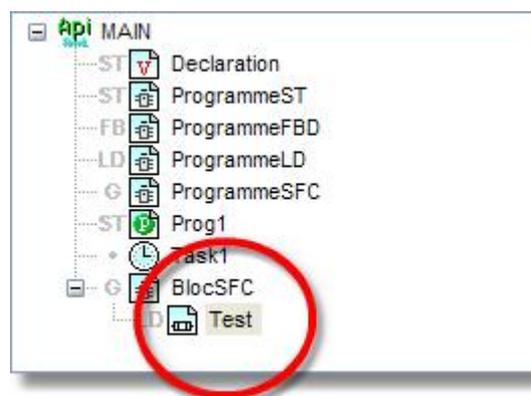


The button surrounded (⏏) permits to jump to the definition of the external action. If this one already exists, it is displayed directly. Note that this action call can make reference to an action named and defined "in situ", in this case the corresponding part of the diagram is displayed.

If the action doesn't exist yet, a click on the button ⏏ provoke its creation. The following window is first of all displayed, permitting to define the language of definition of the action :



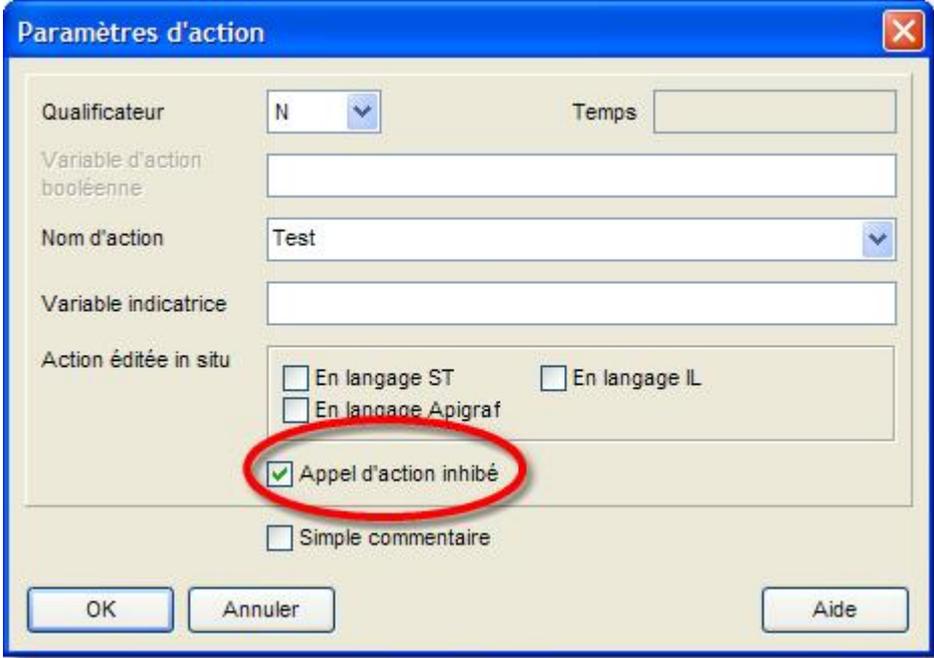
Once the window validated, the action is added to the actions of the current POU (here BlocSFC) :



2.3.2.6.3.5 Inhibition of an action call

It is possible to deactivate the call to an action, without suppressing it completely. This possibility permits to do some tests easily in debug phase.

In the window of action parameterization, check the box "Call of action inhibited". The action is not compiled and appears in grey in the graph.

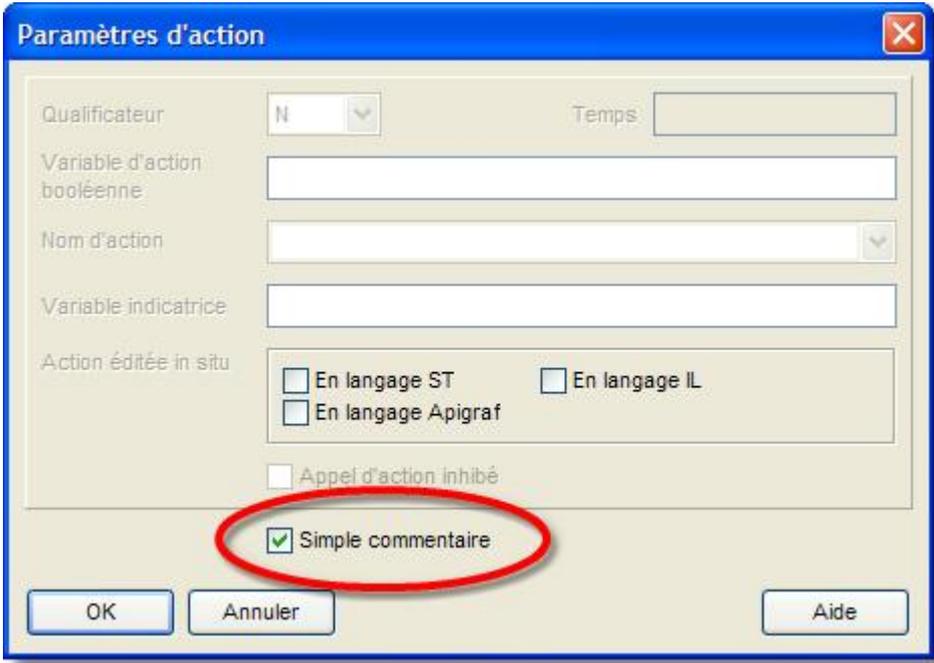


The screenshot shows the 'Paramètres d'action' dialog box. The 'Qualificateur' is set to 'N'. The 'Nom d'action' is 'Test'. The 'Appel d'action inhibé' checkbox is checked and circled in red. Other options like 'En langage ST', 'En langage IL', and 'Simple commentaire' are unchecked.

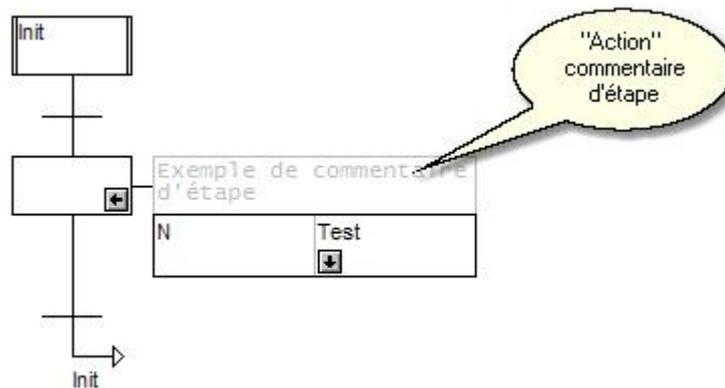
2.3.2.6.3.6 Create a step commentary

To comment the step or the actions of this one, a commentary can be defined with an action parameterized as commentary.

For it, check the option "Simple commentary" in the window of action parameterization. The commentary can be entered then directly in the graph to the manner of the instructions in situ.



The screenshot shows the 'Paramètres d'action' dialog box. The 'Simple commentaire' checkbox is checked and circled in red. Other options like 'Appel d'action inhibé', 'En langage ST', 'En langage IL', and 'Simple commentaire' are unchecked.



2.3.2.6.3.7 Manipulation of the actions

The following manipulations are possible on the actions

Add an action	Right click on the step then on "Add an action"
Suppress an action	Right click on the action then on "Suppress the action"
Move up an action	Right click on the action then on "Move up the action"
Move down an action	Right click on the action then on "Move down action"
Parameterize an action	Double click on the action, or right click on the action then on "Parameterize the action"
Copy an action	Buttons to copy/paste, or to drag-and-drop with the mouse
Displace an action	Buttons cut/paste

2.3.2.6.4 Copy of the elements of a SFC diagram

The steps, the actions and the transitions can be copied then pasted in a same SFC diagram or in another SFC diagram.

The selection of the elements to copy is made by a simple click on the object or by a selection rectangle (definite with the mouse).

The copy can be made then by clicking on the button "Copy" () of the toolbar.

To recopy the selection, it is sufficient to select a step, an action or a transition (of the same SFC diagram or another) and to click on the button to "paste" ().

It is also possible to use the drag-and-drop method : select the objects, place the mouse on the selection, press the left key of the mouse and maintain it while moving the mouse until the object before which the selection must be recopied.

In order to respect the syntax of the diagram, the copy is done in an intelligent way in the following cases :

- copy of a step before a step (or an action) : a transition is automatically added between the two steps
- copy of a transition before another transition: an empty step is automatically added between the two transitions
- paste an action is only possible if the selected element for recopies is a step
- recopy of an initial step or a return toward a step is forbidden
- recopy of steps or transitions before the initial step is forbidden

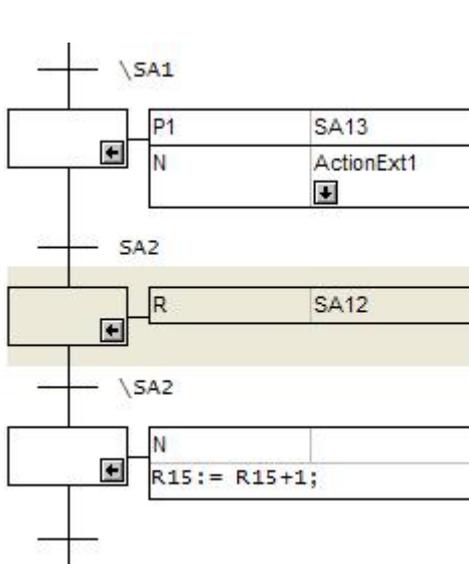
2.3.2.6.5 Deletion of elements of a SFC diagram

To suppress elements of the diagram, select them then to hit the Delete key (Del).

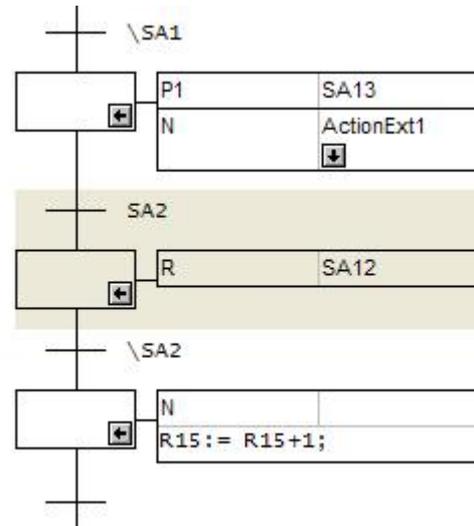
It is also possible to make a deletion with copy of the suppressed elements in the clipboard, with the button "cut" ().

Attention however, the deletion is not always possible, because the following principle must always be respected: after every operation, creation, copy or deletion, the diagram must be always in a valid state, that is to say that the rules of disposition of the steps and transitions must be observed.

For example, it is not possible to suppress a step only. It is necessary to suppress a transition also, before or after, so that the diagram remained correct.



Impossible deletion



Possible deletion

2.3.2.6.6 Example of creation of a SFC diagram

The example is a chaser containing 3 lamps represented by the variables Lamp1, Lamp2 and Lamp3.

The starting of the chaser is made according to the variable StartChaser and stop when the variable StopChaser is "true".

Select (or create) a POU of type Function block "BlocSFC" in SFC language.

The first stage to do is to create the variables of the equation.

Select the tab "Declarations" and insert the following declarations :

```

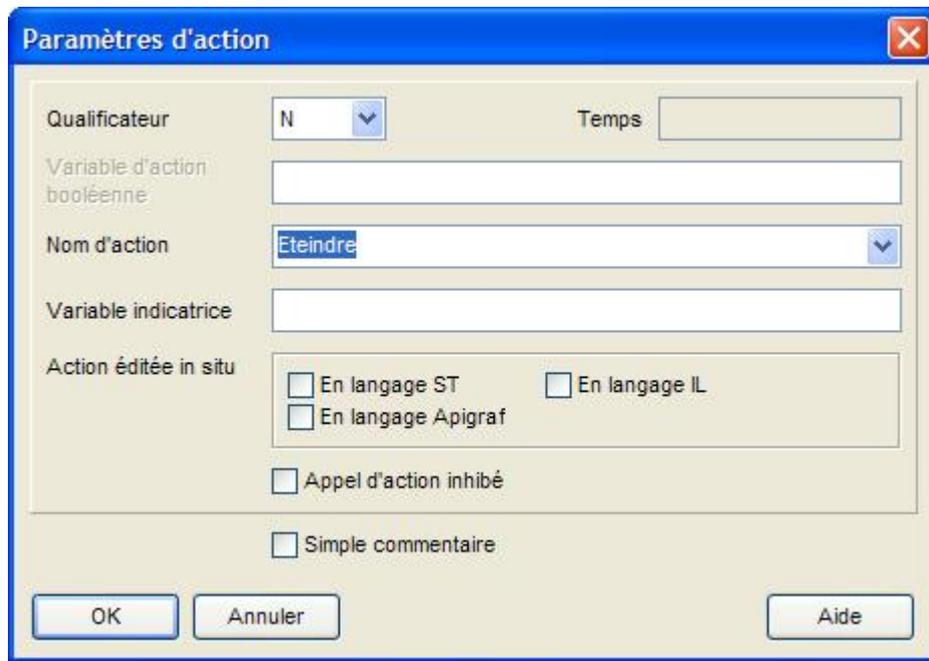
Var_Input
  StartChaser : bool;
  StopChaser  : bool;
End_Var

Var_Output
  Lamp1      : bool;
  Lamp2      : bool;
  Lamp3      : bool;
End_Var

```

Select the tab "Instructions."

Select the "Init" step then use the contextual menu to add an action with the N qualifier (action executed as long as the step is active) and name it "Extinguish".



Click on the button  and choose the ST language.

A POU named "Extinguish" is created then in the tree of the library, the tab page "Instructions" that is associated to it is automatically selected.

Write the following instructions :

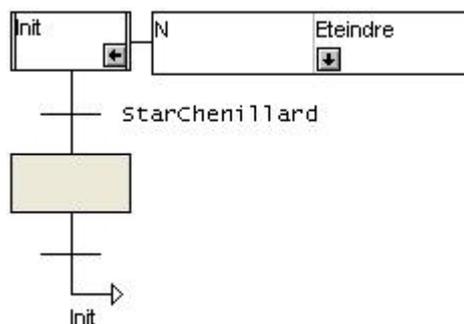
```
Lamp1: = false;
Lamp2: = false;
Lamp3: = false;
```

Re-select the POU "BlocSFC" in the tree of the library.

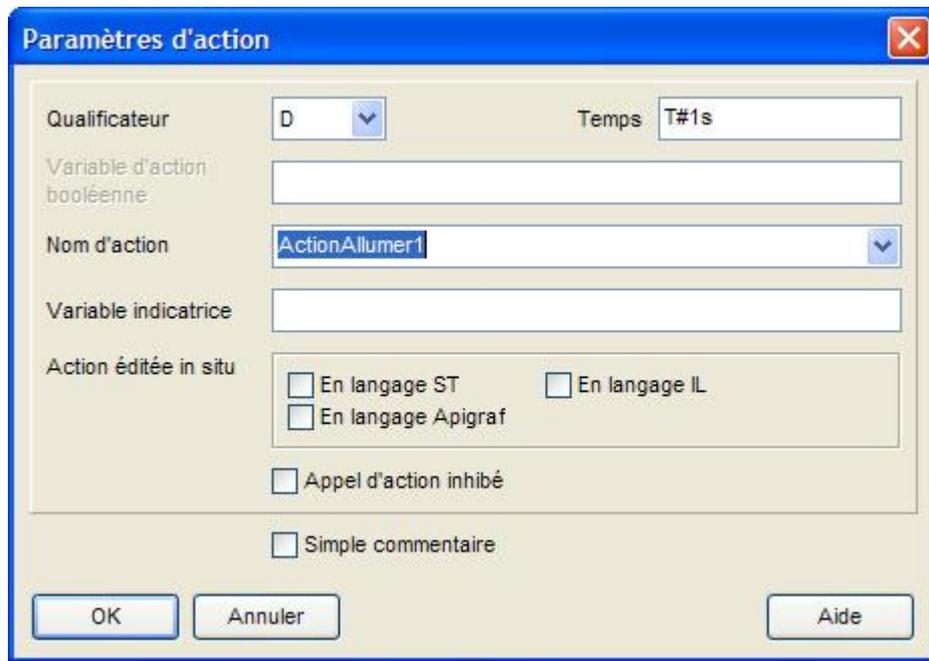


Select the transition of the diagram, and use the contextual menu (right click of the mouse) to enter the transition: "StartChaser."

Re-select the transition then click on  to create a new step.



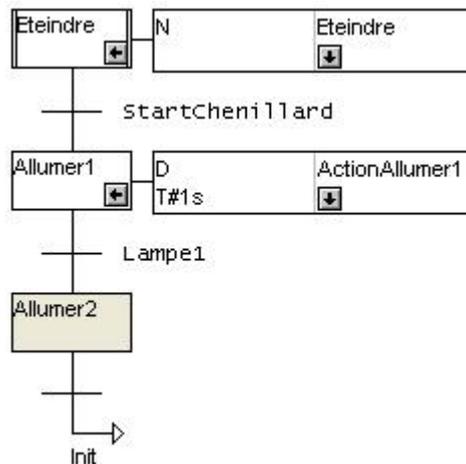
Use the contextual menu to rename the step "Lighting1" to add an action with the qualifier "D", a time "T#1s" (the action executes 1 second after the activation of the step) and name it "ActionLighting1."



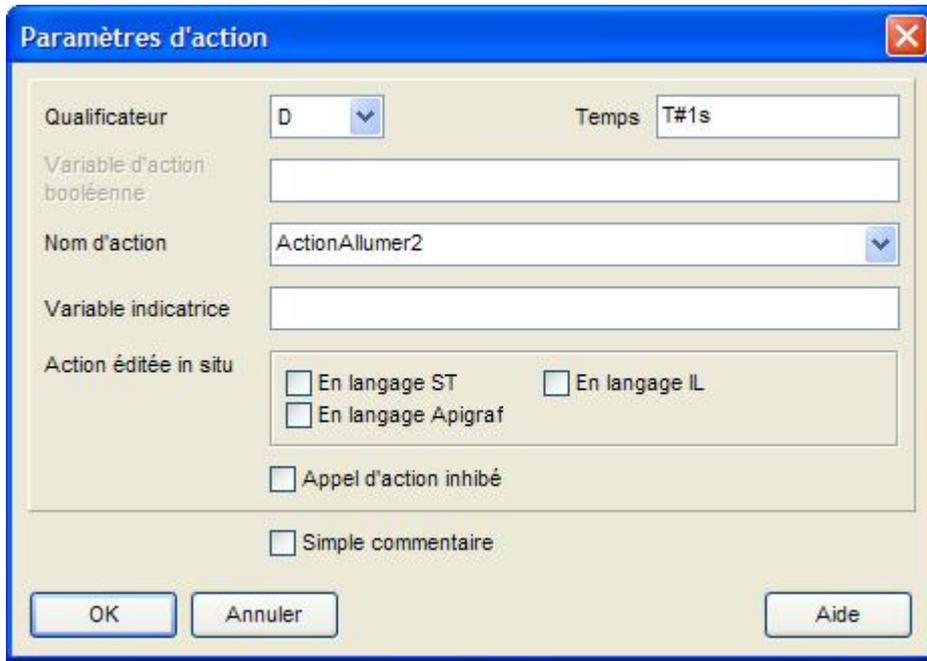
Click on the button  and choose the ST language.
 A POU named "ActionLighting1" is created then.
 Write the following instructions :

```
Lamp1: = true;
Lamp2: = false;
Lamp3: = false;
```

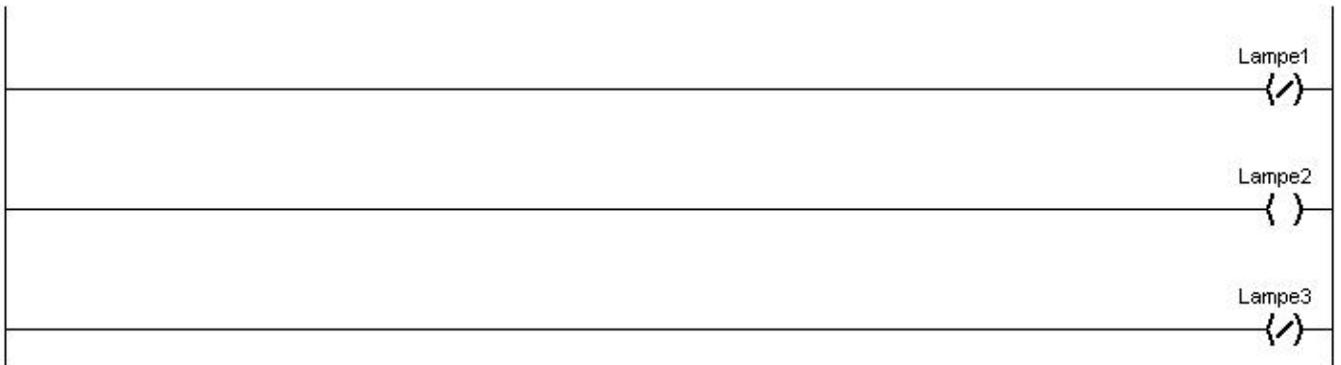
Re-select the POU "BlocSFC" in the tree of the library.
 Select the following transition, there add the "Lamp1" condition, re-select it then click on  to create a new step.



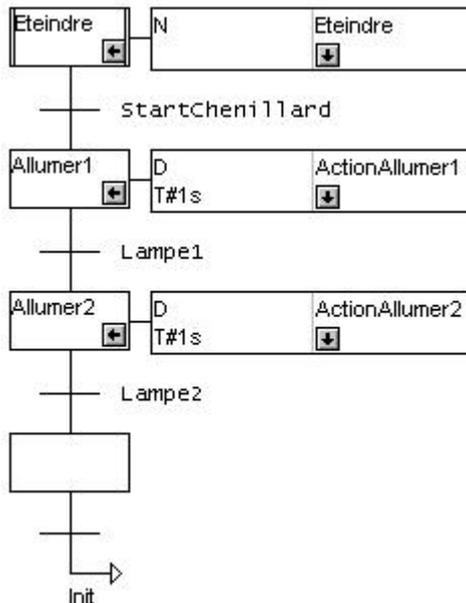
Use the contextual menu to rename the step "Lighting2" then add an action with the "D" qualifier, a time "T#1s" (the action executes 1 second after the activation of the step) and name it "ActionLighting2."



Click on the button  and choose the ladder language.
 A POU named "ActionLighting2" is created then.
 Add the diagrams to achieve the following equations :



Re-select the POU "BlocSFC" in the tree of the library.
 Select the following transition, there add the "Lamp2" condition, re-select it then click on  to create a new step.

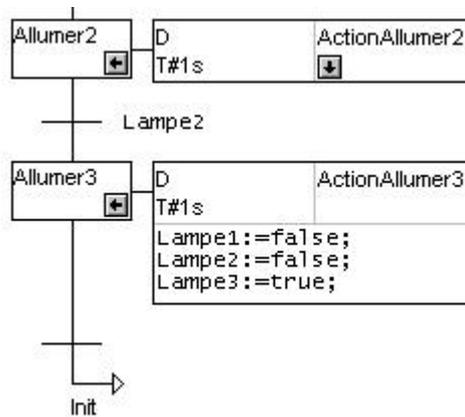


Name the new "Lighting3" step and create a "ActionLighting3" action ("D" qualifier, Time of 1 second) by checking

the option "Edition of the action in situ" in ST.

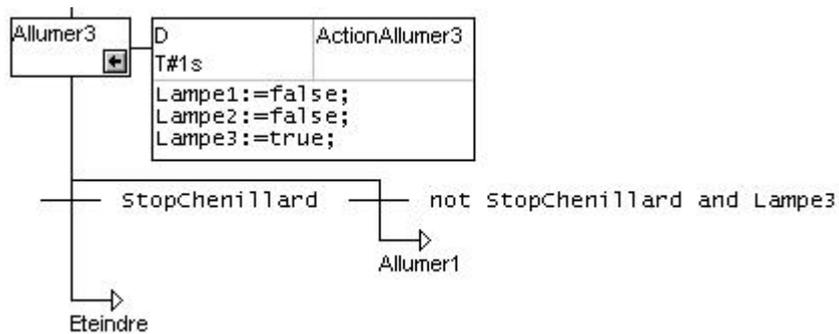
Select the rectangle dedicated to the instructions to write the following instructions then :

```
Lamp1: = false;
Lamp2: = false;
Lamp3: = true;
```



Select the following transition, there add the "StopChaser" condition, re-select it then click on to create a divergence OR.

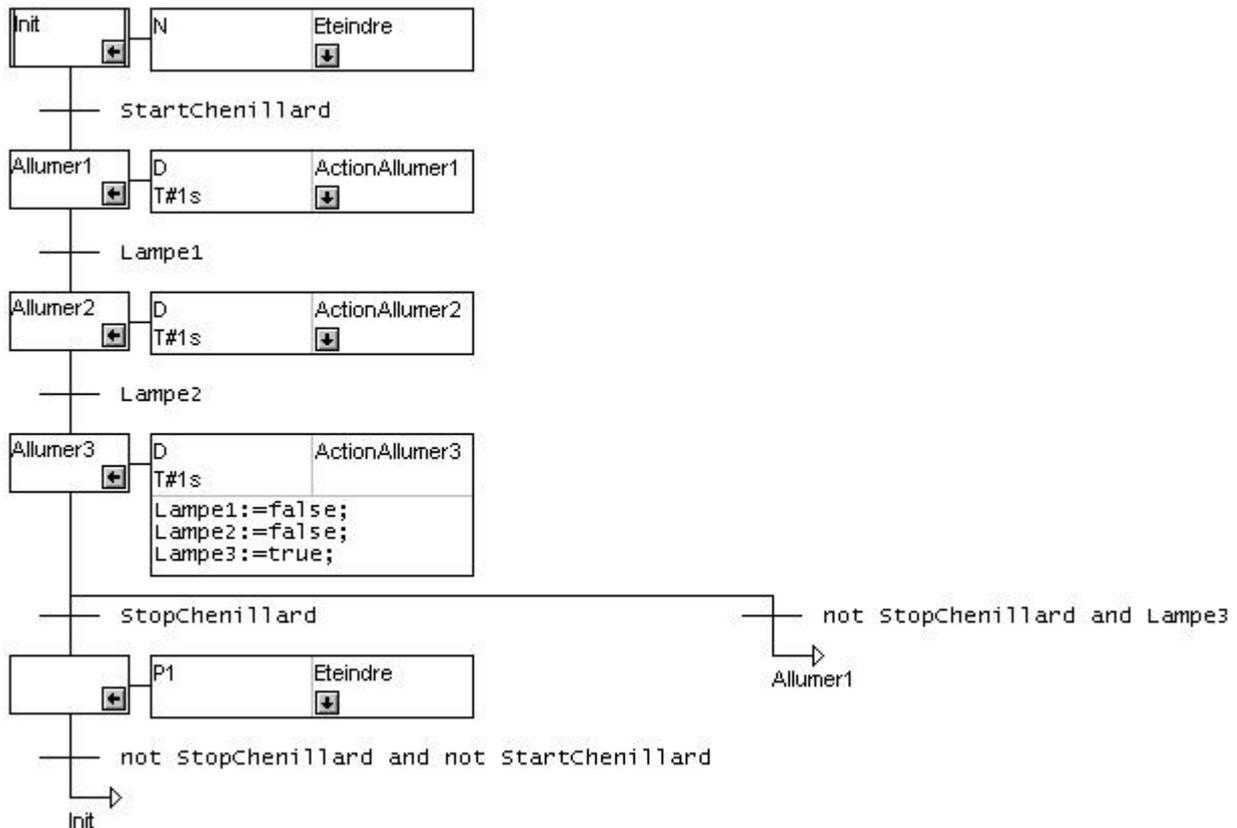
Add the condition "not StopChaser and Lampe3", re-select this parallel transition then click on by specifying the step back "Lighting1."



Select the "StopChaser" transition then click on to create a new step.

Add the action "Extinguish" (to call the previously created action) with the P1 qualifier (execution at the activation of the step).

Select the last transition and add the condition "not StopChaser and not StartChaser".



The diagram is complete.

To verify that no error has been committed during the creation, compile it.

To test it, select a block of program (Prog1 in the example), declare a variable of "BlocSFC" type and call the block in the instruction part :

Declaration :

```
VarTestSFC: BlocSFC;
```

Instruction :

```
VarTestSFC (StartChaser: = SA0, StopChaser: = SA1, Lamp1 = > L2, Lamp2 = > L3, Lamp3 = > L4);
```

Then compile to execute.

2.3.2.6.7 Reset of a SFC diagram

It can be necessary to force a SFC diagram to its initial step.

This operation doesn't pose a problem in the diagram itself if this is necessary in some particular steps, it is then a simple return to the initial step after a transition.

If on the other hand the forcing must be done on a condition whatever is the active step, it is preferable to program it from the outside of the diagram, for example in the block from which the diagram is called. The forcing is done with the "ResetSFC" function.

If for example the diagram is programmed in a block with instance variable named MainSequence, the reset is gotten by making :

```
ResetSFC (MainSequence)
```

Attention, if this diagram also contains other sequences, in sub-blocks or actions, all SFC sequences are forced also to their initial step.

2.3.2.7 Creation of ST block, IL and textual SFC

The creation of blocks programmed in one of these textual languages is made as for the graphic languages. On the other hand the edition of the code only uses a classic text editor.

2.3.2.8 Manipulation of the POUs of a library

The POUs can be displaced and copied from a group to the other, from a library to another, by dragging them with the mouse.

They can be moved up or down in the conception tree with the buttons  and .

To rename a POU, make a right click and use the option "rename". The workshop verifies that the name that you give to a POU is not already used in the loaded libraries.

2.3.3 Printing

Several levels of printing are proposed :

- printing of the diagrams only for the graphic languages (SFC, LD and FBD)
- printing of a POU (information, declarations and instructions)
- printing of the user library
- printing of a project (all user libraries)

2.3.3.1 printing of diagram

In the case of the graphic languages, it is possible to print the diagrams directly in the tab page "Instructions" by clicking on the icon  of the toolbar.

2.3.3.2 Printing of POU

The option "Print" is available in the popup menu of the POUs.

The printing of the content of a POU presents itself on one or several pages and include :

- the available information on the POU among the following :
 - the properties
 - the symbol interfaces
 - the descriptive of use
 - the declarations
 - the instructions
- the information on the POUs belonging to the selected POU (case of the POUs belonging to subgroups, or actions defined in a POU in SFC language)

2.3.3.3 Printing of a library

The option "Print" is available in the popup menu of the libraries.

It provokes the printing of all POUs contained in the library.

2.3.3.4 Printing of a project

The option "Print" is available in the "Project" menu of the programming workshop.

It provokes the printing of all libraries opened in the project.

3 Supervisory workshop

For a first starting, to go to consult the chapter :
[Creation of supervisory forms - general Principles](#) 

3.1 Main menu of the application

3.1.1 Menu file

3.1.1.1 Menu file - New

The menu " new " is used to create a new supervisory form. Once created, the supervisory form must be saved.

3.1.1.2 Menu file - Select project

The menu " projects " call the project's selection page. This menu permits to select a project ,to create a new project or to suppress a project.

This menu is not available if the supervisory workshop has been started since Optima PLC.

3.1.1.3 Menu file - Save

The menu " Save " permits to save the current supervisory form in the project. If the current supervisory form has just been created, the user will be asked to enter a file name to save the form.

3.1.1.4 Menu file - Open

The menu " to open " permits to open a supervisory form of the project in the supervisory workshop in designing mode.

3.1.1.5 Menu file - Record as

The menu " Save as " permits to save an existing supervisory form under another file name. The user is asked to enter the new file name to record the form. It is often used to create a copy an existing form.

3.1.1.6 Menu file - Quit

This menu permits to quit the workshop.

3.1.2 Menu Edit

3.1.2.1 Menu edition - Undo

Undo the last action.

3.1.2.2 Menu edition - Cut

Erase all selected objects and places them the windows clipboard. These objects will be able to be " glued " thereafter.

3.1.2.3 Menu edition - Copy

Copy all objects selected in the clipboard

3.1.2.4 Menu edition - Paste

Paste all the objects previously copied in the clipboard on the current form.

3.1.2.5 Menu edition - Delete

Delete all selected objects on the current form.

3.1.2.6 Menu edition - Select All

Select all existing objects on the current form

3.1.3 Menu Tools

3.1.3.1 Menu Create Shortcuts

The user can create some shortcuts to supervisory forms and place these shortcuts on the desktop, in the programs menu or in the automatic starting of WINDOWS®

The menu opens the following window :



The different controls of this page are :

Button "select"

Select the supervisory form on which the shortcut is created.

Shortcut's Name

Enter here the shortcut's name. The default name is : " Shortcut to " + the supervisory form's name. This name will appear at the bottom of the shortcut

Shortcut destination

This control is composed of three checkboxes, the user can specify the destination of the created shortcut (Desktop, program's menu or Windows starting). It is possible to select several destinations simultaneously.

Selection of the icon

It's possible to select the shortcut's icon by clicking on the button " Icon " and choosing a file valid icon (* .ICO).

The default icon is the one of the supervisory workshop.

A preview of the icon selected is also available in the window.

3.1.3.2 Menu User translations - Open a language file

The user can manage several languages in the supervisory forms (to see translation's management). This menu opens an existing file in the project (the name of the language file is like : langage.1 langage.2 langage.3.)

3.1.3.3 Menu User translations - New language file

Create a new language file. If no language file already exists in the project, a file LANGAGE.1 is created. If a file of language already exists (LANGAGE.1), then a new file (LANGAGE.2) is created, and so one.

3.1.3.4 Menu User translations - AutoExtract page texts

Extract automatically all texts on the current supervisory form to put them in a language file. The user can do this operation on several language files and then make his own translation in these language files. He also has to decide which number correspond to which language.

Example : the original language is English. When the supervisory forms all done, the user can extract the texts to both files LANGAGE.1 and then LANGAGE.2.

LANGAGE.1 will contain the original texts (In English)

The user can translate texts in file LANGAGE.2 (In French for example)

The user will have two languages defined (1 for english and 2 for french). He can then use the local macros CHNUSERLANGUAGE(NumLanguage) to change dynamically the current language of the supervisory forms.

The menu opens the following window :

The name of the file specifies the file language that is going to be used to do the operation. If this one doesn't exist, it will automatically be created.

The possible operations are :

Replace Text by @ LGx: Text

With this operation all text fields of all objects of the supervisory form will be replaced by " @ LGx : " followed by the initial text , x representing a number given automatically to reference the translation.

Example : If a page contains a button windows whose text is " Quit " and if the LANGAGE.1 file contains :

@ LG0 :

New

@ LG1 :

Page2

The text of the button windows will be replaced by " @ LG2: Quit " and the LANGAGE.1 file will become :

@ LG0 :

New

@ LG1 :

Page2

@ LG2 :

Quit

Replace Text by @ LGx

This operation is the same that the previous to the difference near that the texts of the objects won't contain their initial value in commentary.

" To leave " would be replaced by " @ LG2 " in the example of the over.

Replace @ LGx by the chains of the file

This operation is the inverse operation of the two previous operations.

If the texts of a supervisory form have been extracted in a file language, this operation permits to recover the initial texts of the objects in the mentioned file language and to reintegrate them in the objects of the page.

Option " Use already referenced texts "

This option permits to search for first if the text to extract is present in the file language before creating a new reference. If it is the case, the existing reference will be used.

3.1.3.5 Menu Apigraf IP Explorer

Launch the APIGRAF IP- OPTIMA PLC Explorer. This software is a file explorer that can explore some distant APIGRAF IP Workstation (running under DOS) and also offers the possibility to manage or to transfer some files on DOS PLC or either WIN stations provided of the APIFTPServer application.

3.1.3.6 Menu Alarms Manager

Launch the Alarm Manager. This application can define and manage general alarms on the system. Alarms can then be visualized in a supervisory form or in another distant Alarm Manager setup in slave mode.

3.1.4 Menu Display

3.1.4.1 Menu Display - ToolBar

This menu references the four available tool bars.

Main Menu :

If this menu is checked, the "Main menu" toolbar will be visible.

Edition :

If this menu is checked, the "Edition" toolbar will be visible.

Function :

If this menu is checked, the "Function" toolbar will be visible

Palette :

If this menu is checked, the "Palette" toolbar will be visible. The "Palette" toolbar contains all available supervisory objects.

3.1.4.2 Menu Display - Function panel

If this menu is checked, the "Function" panel will be visible. The small window "Functions" appears at the bottom of the screen. This window permits to manage some functions like : Execute a supervisory form, get the form properties, resize an object, duplicate an object as a network,..

3.1.4.3 Menu Display - Object's explorer

If this menu is checked of the object's explorer will be visible.

The object's explorer list in a treeview all present objects in the current supervisory form. It's possible in this explorer to force an object to be invisible during the phase of conception of the supervisory form.

More details in the following chapter :

[Object's Explorer](#)^[290]

3.1.4.4 Menu Display - Alignment panel

If this menu is checked the Alignment tools window will be visible. The window "Alignment" appears at the bottom of the screen. This window permits to have access to functions of alignment of the objects of supervisory.



The buttons of function of the palette alignment are active or not depending on the number of selected objects in the current supervisory form.

For the four functions on the top, at least two objects selected on the page are necessary and it is necessary that these objects have the same owner: (example: 2 objects in the same panel or 2 objects directly on the same page)

For the four functions of the low, at least two objects selected on the page (can imports the owners of the objects) are necessary.

Select several objects :

To click on the first object to select. Then push on the SHIFT key of the keyboard and maintain this key ON.

Then click on the other object to select. If an object has been selected by error, click again on this one to unselect it.

Loosen the SHIFT key once the selection is complete.

3.1.5 Menu Object

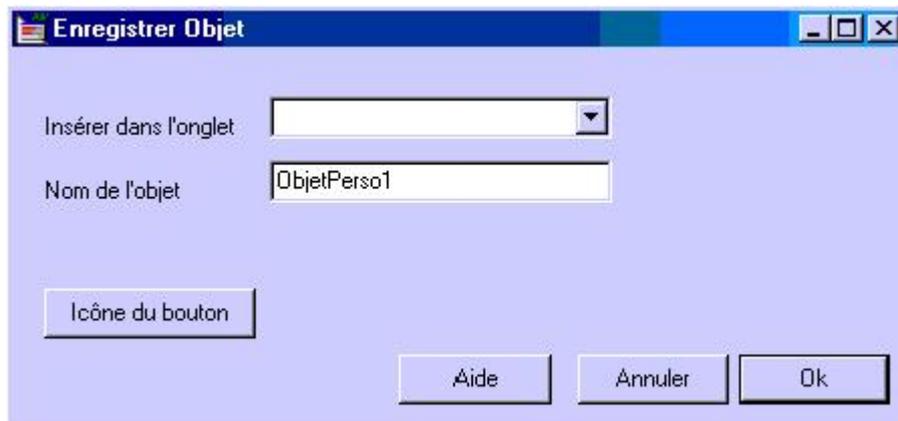
3.1.5.1 Menu Object - to Add

The user can record a tuned object with all its parameters.

Make a right click on the object to make appear the popup menu and then click on "Record object".

A button will be created in a new tab or an existing one. This button allows to recall the saved object.

The menu opens the following window :



The different parameters are:

Insert in tab:

This combo box list all existing tabs on the object tool panel. The user can select an existing tab or to enter a new tab name in the input box. In this case a new tab with the entered name will be automatically created.

Name of the object:

The user can assign a name to the tuned object. This is the name of the object selection button that will be displayed in the Hint in when the mouse cursor will pass above. This name is also the file name of the recorded object.

(ex: User0ObjetPerso0.bib)

Selection button icon :

The button will have an icon by default:

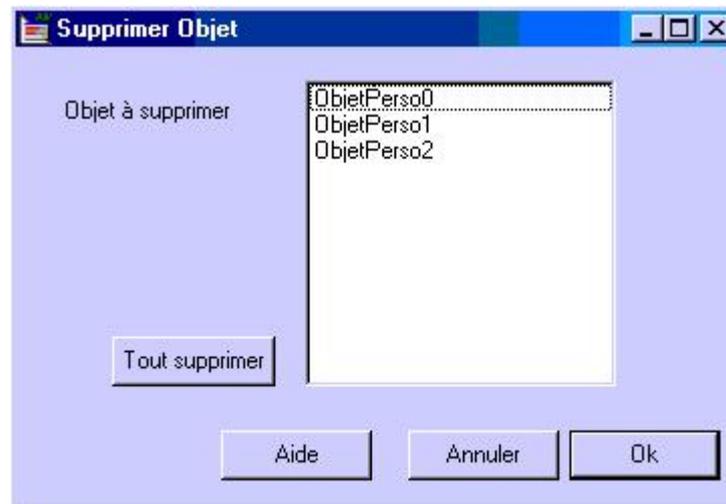


The user can personalize the button icon by clicking on the button and by choosing a valid icon file.

3.1.5.2 Menu Object - to Suppress

This menu permits to erase the personalized objects. It will hide the button, the created tab doesn't have a button more in as well as the file of the object to record.

This menu opens the following window :



The paramêtres is they according to:

Object to suppress:

This list relates all objects that have been recorded (objects personalized).

It is sufficient to select the or the objects to suppress. To select several objects, to click on an object and made slip the mouse until the last object that you want to suppress or to click on those that you want while remaining to push on the "Control" key.

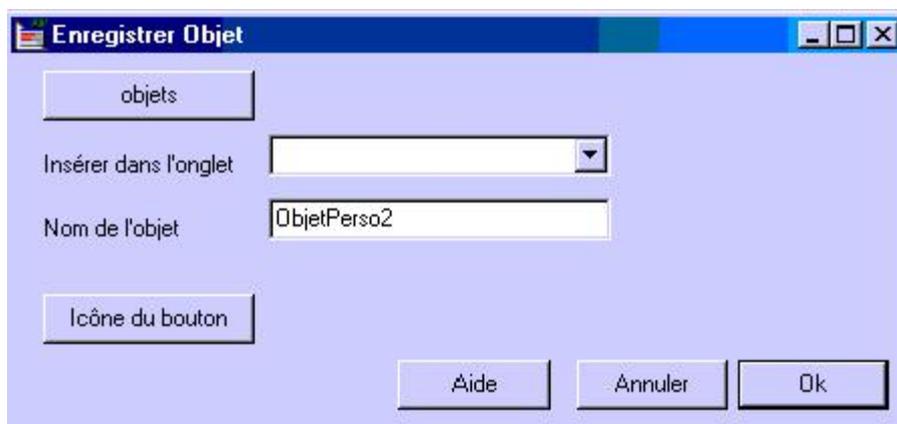
All to select:

Permits to select all objects of the list.

3.1.5.3 Menu Object - to Import

This menu allow the user to import tuned objects saved in a file with the "*.bib" extension. An object selection button will be created in the defined tab.

This menu opens the following window :



The different parameters are:

Objects:

Open a file dialog to select an object file to import. This file must have the "*.BIB" extension.

Insert in the tab:

Select here an existing tab in the object tool panel to create the object selection button. The user can also enter in the input box a new tab name. The corresponding tab will be automatically created in the tool panel.

Object name:

Defines the object name. The user can specify a new name for the imported object or keep the default name of the imported object.

(ex: User0ObjetPerso0.bib)

Button icon:

The objet selection button created in the tool panel will have the default icon :



The user can choose a valid icon file in the file dialog for the object section button.

3.1.6 Menu Macros

3.1.6.1 Menu Macros - local Macros

This menu displays the list of the available local macros.

The user can execute one local macro from the list as follows :

1. Select in the list the macro to execute by clicking with the mouse on the macro line. This one appears selected.
2. Click on the button "to execute "
3. A window containing the text of the macro appears to the screen.
4. Complete the text of the macro if needed (For example : Add the parameters in brackets if the macro need some parameters)
5. Validate with the "OK" button to execute the macro.

3.1.6.2 Menu Macros - Macros PLC

This menu is used to execute some macros on a local or distant PLC. (OPTIMA PLC or APIGRAF)

The window displayed on screen allows to enter the IP address of the OPTIMA PLC or APIGRAF PLC workstation (127.0.0.1 if OPTIMA PLC runtime is on the local workstation) and the text of the macro to execute. Click on the button "Execute " to execute the macro on the local or the distant station.

Note : The field of the IP address of the PLC workstation is automatically completed by default with the default address distant IP defined in the project options.

3.1.6.3 Menu Macros - Macros User

The user can create special DLL files (libraries) with special functions that can executed in the supervision. The macros user are of the macros contained in DLL developed outside of the workshop (in C++, pascal and so one).

This menu is composed with 3 sub menus :

User DLL List :

This submenu list the user DLLs user that have been loaded in the project.

Load a user DLL :

Load a selected user DLL. The DLL is selected in a file dialog. The DLL will be reloaded automatically to the execution of the project.

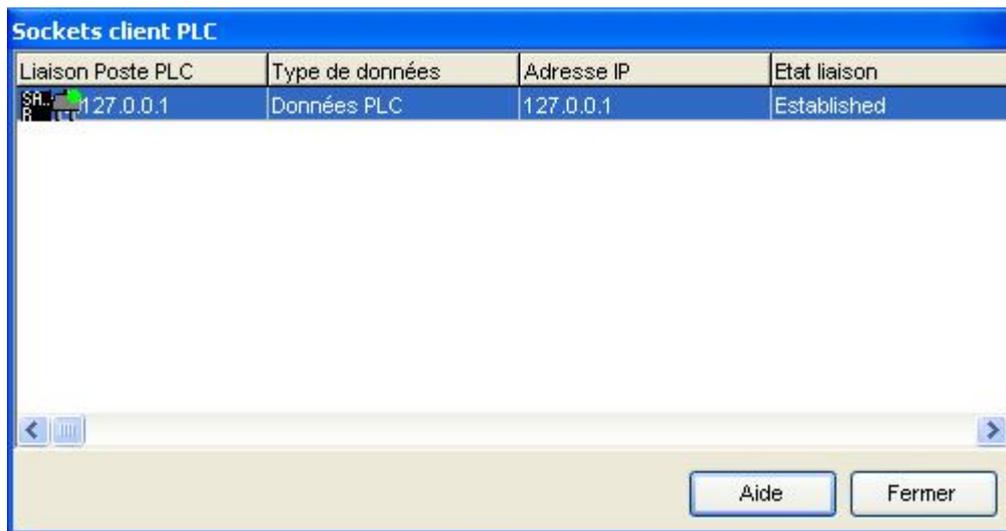
Unload a user DLL :

Unload the selected DLL usert. Once unloaded, the DLL won't be loaded anymore at the time of a new opening or execution of the project.

3.1.7 Menu Diagnoses

3.1.7.1 TCP / IP PLC data client

This menu opens the diagnosis window of the data exchange sockets with OPTIMA PLC or APIGRAF.



This window lists all existing sockets connected or not to runtimes OPTIMA PLC or APIGRAF.

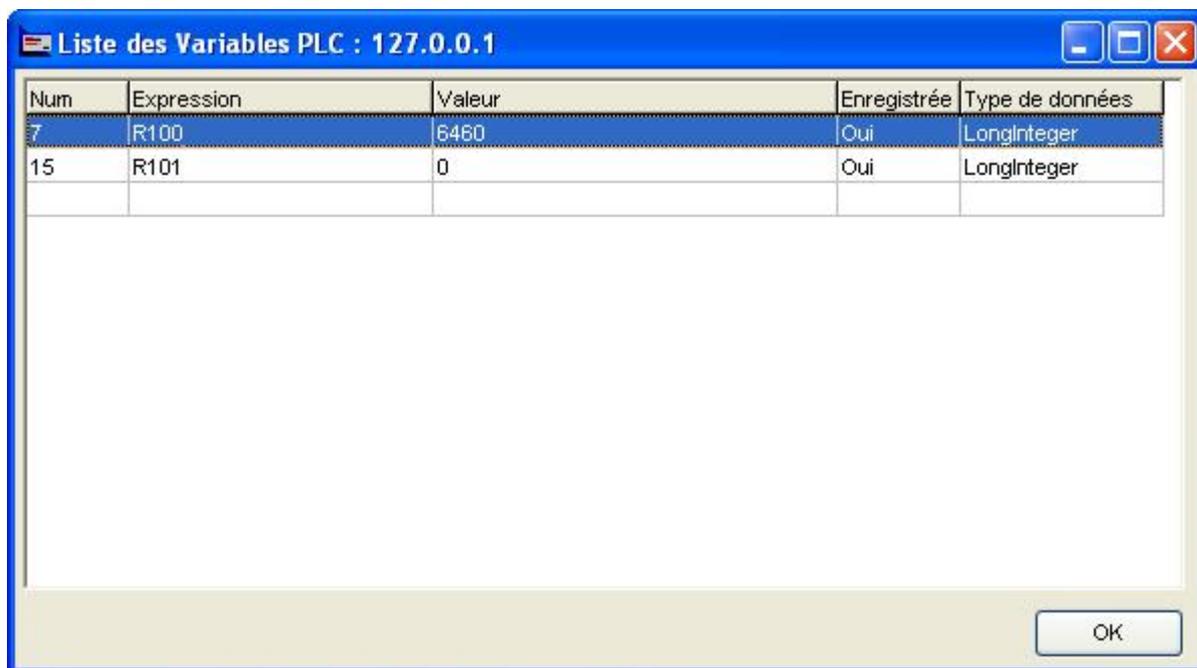
The IP address 127.0.0.1 always corresponds to the local address of the PC.

For every socket is mentioned the distant IP address and the connection state.

The socket is connected when the state is "Established"

When distant samples tables are also used in the supervisory, the sockets of links corresponding to the different distant samples tables appears in this list.

For the data exchange sockets (see the field data type in the list) the user can get some details by doing a double-click on the socket in question. The following window opens then :



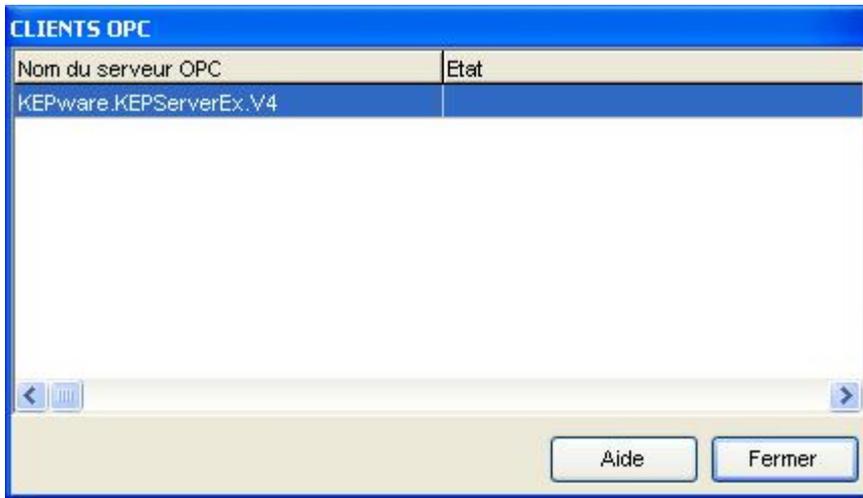
This window lists all variables exchanged on the socket (here R100 and R101) as well as :

- The variable value (if the one is known therefore on the distant station)
- If the value is recorded or not (if it is known on the PLC).
- The type of data of the variable if it is recorded.

3.1.7.2 OPC clients

This menu opens diagnosis window of the links with OPC servers.

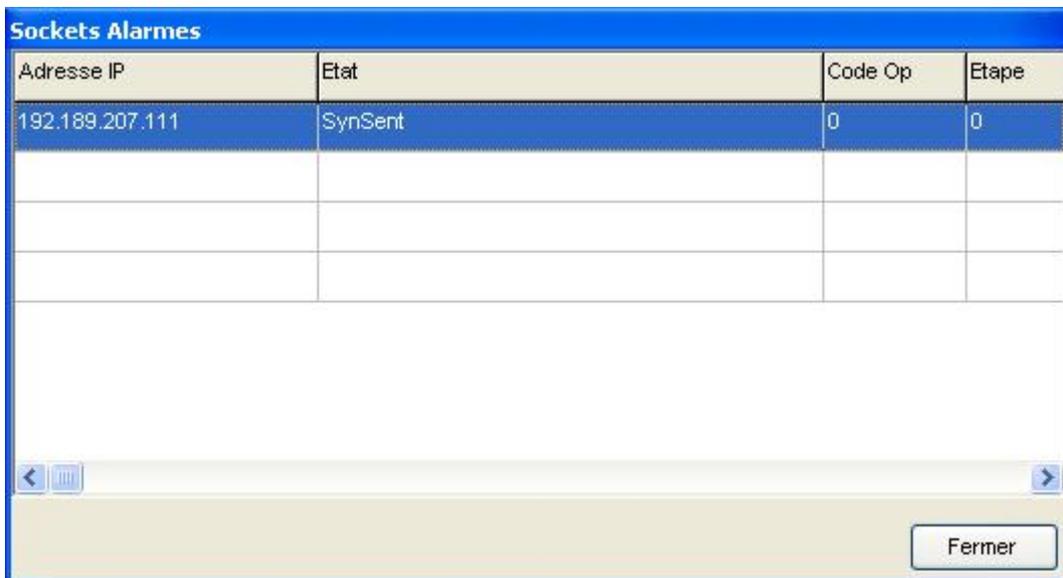
This window lists the OPC servers with which a link is asked.



3.1.7.3 TCP / IP Alarms data clients

This menu opens the diagnosis window of the alarm data exchange sockets. These are used to share the alarms centralized by Optim'Alarm software.

The supervisory object "Alarm history" use one socket of this type.



The window lists the links asked for the update of the alarms history.

3.1.7.4 Measure samples tab linked in memory

This menu opens the diagnosis window of the memory linked files that contains the local samples tables coming from optima PLC.

When a measure samples table is supervised in an object Graph coming from Optima PLC on the local station (with IP adress 127.0.0.1), the datas are exchanged in memory and no with a TCP/IP socket as it is the case for tables coming from Optima PLC on distant station (on the network ethernet) or from APIGRAF

TABLEAUX DE MESURES LIES EN MEMOIRE					
Nom Tableau	Nom Fichier mappé	Taille	Nb Acquisitions	Version données	Nb User
TABCANAL1	_TSOPLC_TABCANAL1	4000	3999	22699	1
TABCANAL0	_TSOPLC_TABCANAL0	4000	3999	22699	1

Réinitialisation des liens tableaux

Fermer

The window lists the measure samples tables linked in memory with :

- Name of the table: the one that is entered in the object of supervisory graph
- Name of the file memory mapped in memory : This name is automatically created with the table name
- Size: maximum number of samples in the table
- Samples number: number of present samples in the table
- Data version : version number of the last data updates
- Nb User: number of demand of supervisory on this table. This number is incremented every time an object graph is linked to this table.

If the link has not been established during the supervisory starting, some data are marked with "NC" as the fields : Size, Samples number.

3.1.8 Menu Options

3.1.8.1 Menu Options - general Options

The menu " General options " open the following window :

Options du projet

C:\PROGRA~1\OPTIMA~1\Projets

Commentaires :

Version :

Langage : Français Anglais

Connection APIGRAF PLC

Cette adresse IP est l'adresse par défaut du projet.

Clavier virtuel automatique (utile pour les écrans tactiles)

Commentary :

Define some commentaries to the project.

Version :

The user can associate a version number to his supervisory project.

Language of the application :

The French and English languages are currently available. Select the desired language in the dropped list.

Options defined in this page will be taken in account after clicking on the button " OK " on the page.

Remark : These informations appear in the project's selection window for the current project.

Button " Distant IP Address"

This button opens an input box to enter the IP address of the workstation to supervise. This IP address will be taken when no other IP Address is defined in other levels (Form IP Address, Panel IP Address or Expression IP Address). This address is saved in the INI file of the project (PROJECTNAME.INI).

In the case of an use of the supervisory workshop included in Optima PLC, the IP address to inform is : Local_Host what corresponds at the local station (IP address: 127.0.0.1)

Option "automatic virtual Keyboard"

The virtual keyboard is used on tactile screens that are not fitted with keyboard. It is a keyboard that opens up to the screen when a user enter an input box and that permits to the user to input alphanumeric characters by clicking on the buttons of this keyboard.

This option permits to open the virtual keyboard automatically when the user clicks on an input box.

Example : Alphanumeric virtual keyboard



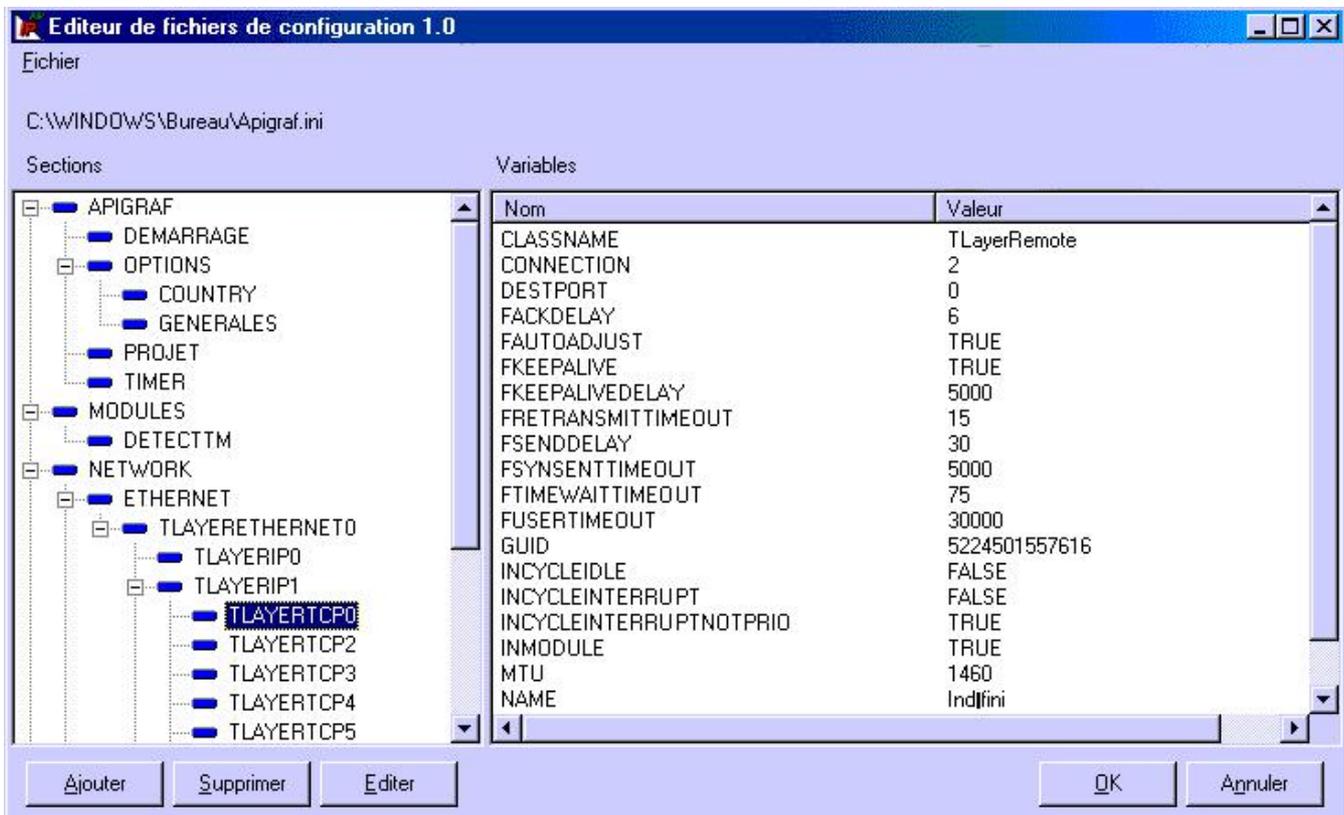
For every input box, it is then possible to define what type of virtual keyboard will be open: alphanumeric or numeric and also to specify if a the input box must or not open this virtual keyboard.

3.1.8.2 Menu Options - Edition configuration

This menu launch the configuration files viewer application : " APINIVIEWER ". This tool allows to visualize and to manage the configuration files coming both from OPTIMA PLC and form APIGRAF.

This menu is not available if the supervisory workshop has been started from Optima PLC.

The main page has the following shape :



The viewer can add or suppress some sections in the configuration files coming both from OPTIMA PLC and from APIGRAF. For the files coming from APIGRAF PLC, it is also possible to add or to suppress some items in the different sections.

3.1.8.3 Menu Options - Management of the users

The Workshop can work with the user management which is common in every OPTIMALOG Software products. This option enables or disables the user management. If it is activated the menu then appears checked

The users management allows to restrict the opening of some supervisory forms to a special user group.

See [Users management](#)⁽³⁰⁶⁾ for more details.

3.1.9 Menu Users

3.1.9.1 To change user

Permits to define himself as the current user under the condition to possess a user name and a password associated.



To inform the name of the user and the password and to validate with the "OK" button.

3.1.9.2 To change the password

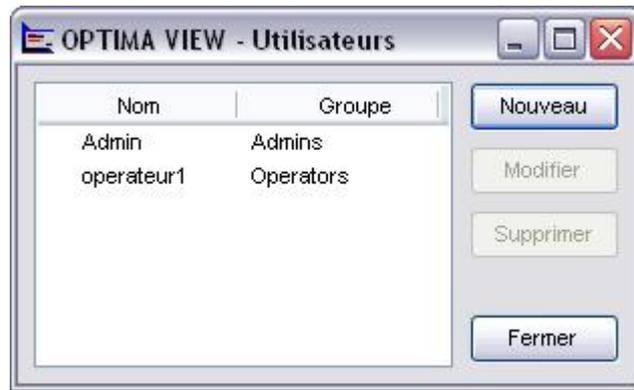
Permits to change the password. The previous password is asked to define the new password.



Attention : the new password must be confirmed and must be entered in the last 2 input fields of the dialog box.

3.1.9.3 List of the users

Permits to consult the list of the users and according to the rights to create or to suppress some users.



3.1.10 Menu Version

Display the information concerning the release number of the software.

This menu is not available if the supervisory workshop is started from Optima PLC.

3.1.11 Menu Help

Open the on line help.

3.2 Creating supervisory forms

3.2.1 General principles

A graphic supervisory form is stocked in a file whose extension is by default (Supervisory form graphic WINDOWS) .PSGW.

Through various graphic objects, a page permits to supervise the expressions that are valued on a OPTIMA PLC runtime, an APIGRAF runtime or on any OPC server.

In the case of an OPC server, user specifies the name of the OPC server on which to connect.

In the case of a workstation OPTIMA PLC or APIGRAF, the user specifies the IP address of the station. For the local workstation, the IP address is : 127.0.0.1

An expression can be a PLC variable, a combination of PLC variables or a logical expression.

Examples of valid expressions :

R2
 R2+R3
 R3 + R4 > R5
 SA2#SA3
 (SA2#SA3) & L10
 .
 .

For every variable, the user can define the name of the OPC server or the IP address of the station OPTIMA PLC or APIGRAF on which will be valued the variable.

So for a variable, one defined an IP address at a time and at a time a name of OPC server then it is the IP address that is going to be taken in account.

Definition of the expression IP address :

These expressions or variables are valued on workstation (OPTIMA PLC or APIGRAF) whose IP address can be defined at different levels :

Project default IP address :

The default IP address of a project can be defined in the menu " OPTIONS " (To see H.3.8)
 This address is taken in account if no IP address is specified at lower levels.

Supervisory form default IP address:

A default IP address can be defined in a supervisory form : (SEE supervisory form properties). This address is taken in account if no IP address is defined at a lower level (In a supervisory object or in the variable expression).

Default IP Address in some controls (panel, pagecontrol, scrolling zone)

Some objects as the panel can contain other objects. In this case, the user can define a default IP address in the object so this IP address will be used by default for all objects in this object. This address is taken in account if no IP address is defined in the expression of the object.

IP Address of an expression :

If an IP address is defined in the expression, this address will be taken in account in every ways.

Example :

R12	Expression of R12
!193.0.1.1 !R12	Expression of R12 valued on a workstation with IP address 193.0.1.1

If the use of the supervisory workshop is joined with Optima PLC, there generally is not a distant workstation to supervise, the IP address to inform by default in the project and in the supervisory forms is :
 Local_Host (what corresponds to the address of the local PC to the address 127.0.0.1) or 127.0.0.1

When a supervisory form is created from Optima PLC, the Local_Host address is automatically entered in the field "IP Default address" of the supervisory forms.

Definition of the expression OPC Server:

These variables are supervised on OPC server whose name can be defined at different levels :

Supervisory form default OPC server name:

If the name of the OPC server by default is defined in a supervisory form (SEE supervisory form properties). This OPC server name is taken in account if no OPC server name is specified at lower levels. (or an IP address).

Default OPC server name in some controls (panel, pagecontrol, scrolling zone)

Some objects as the panel can contain other objects. In this case, the user can define a default OPC server name in the object so this OPC server will be used by default for all objects in this object. This OPC ServerName is taken in account if no OPC Server name is defined in the expression of the object. (or an IP address)

OPC server name defined in the expression :

If the name of an OPC server is defined in the expression, this one will be used to value the expression.

Example :

COMPTEUR1 : Expression of Compteur1
|SERVOPC.PLC|COMPTEUR1 : Expression of Compteur1 supervised on the OPC server: SERVOPC.PLC

3.2.2 Creation of a supervisory form

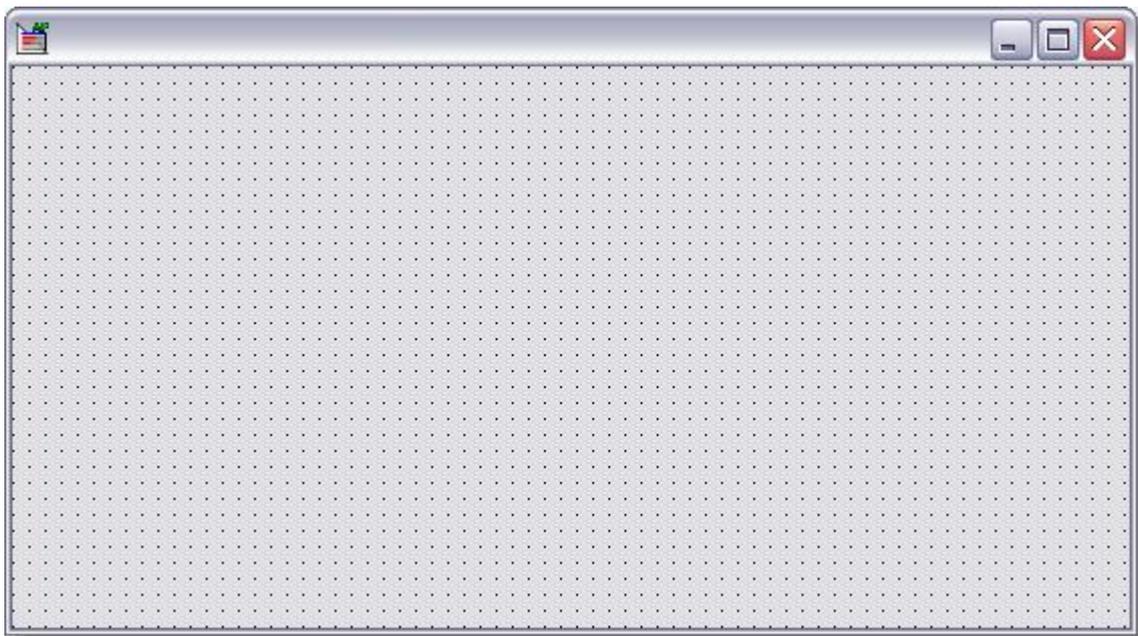
To create a new supervisory form, use the menu " NEW " of the supervisory workshop.

A new window is created, initially empty. Two toolbars are opened at the bottom of the screen. These bars can be moved regardless of the window of supervisory. These bars always stays on top.

The first tool panel is the "alignement panel" which is used to move/align objects on the supervisory form.

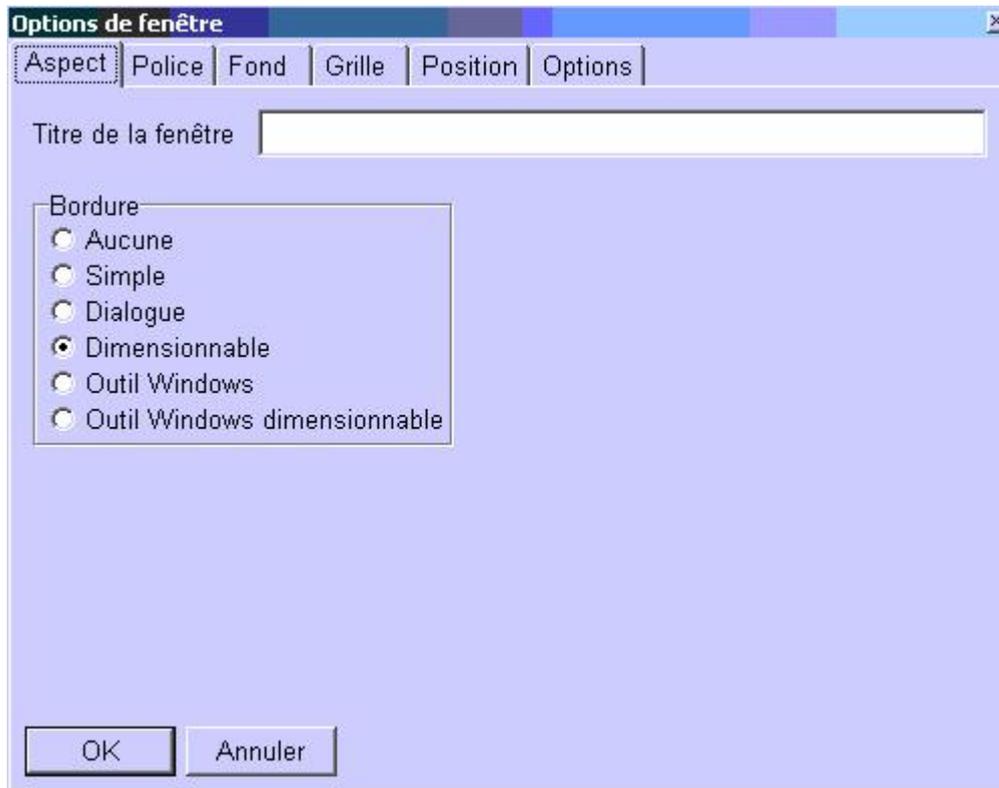
The second tool panel, is the "function panel" . This panel includes some accessible functions since the workshop. This panel shows also the current mouse position (X and Y) on the current supervisory form.

3.2.3 Supervisory form general settings



To define or modify the main setting of a supervisory form, make a right click with the mouse on the form. The supervisory form options window is displayed. This window has many tabs for the options.

3.2.3.1 Style



The user can define the general style of the supervisory form. with the following controls :

Title Window

Enter here the title displayed at the top of the supervisory window. The title will be displayed in execution mode but not in designing mode. In designing mode, the file's name is displayed.

It's possible to put a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with parameters ")

Border

Choose here the general style of the supervisory form border.

Caution : The selected border only appears in execution mode outside of the supervisory workshop.

The different styles are :

No border :

The window has no border and also displays no title.

.

Simple :

The window is standard but is not sizable.

Dialog :

The window looks like of a Windows dialog box. (No sizable, The "Maximize" and "Minimize" buttons are not visible and no icon is displayed on the top of the window.

Sizable :

The window is standard and is sizable with the mouse.

Windows tools :

The window has the appearance of a Windows tool box but is not sizable.

Tools Windows sizable :

The window has the appearance of a Windows toolbar and is sizable.

3.2.3.2 Font



In this tab, the user can define the default font that will be used by default for all objects contained in the supervisory form.

A panel shows a text sample using the current defined font.

To change the default font, click on the button "Default font for objects ", the font dialog box is then displayed on the screen.

3.2.3.3 Background



These options define the supervisory form background

Background color

The user can select the supervisory form background color. This button open the color dialog box.

Picture

The user can specify a picture file that will be displayed on the supervisory form background. The following picture formats are allowed: *.BMP; *.ICO; *.EMF; *.WMF; *.PNG; *.PNG;.....

Delete picture

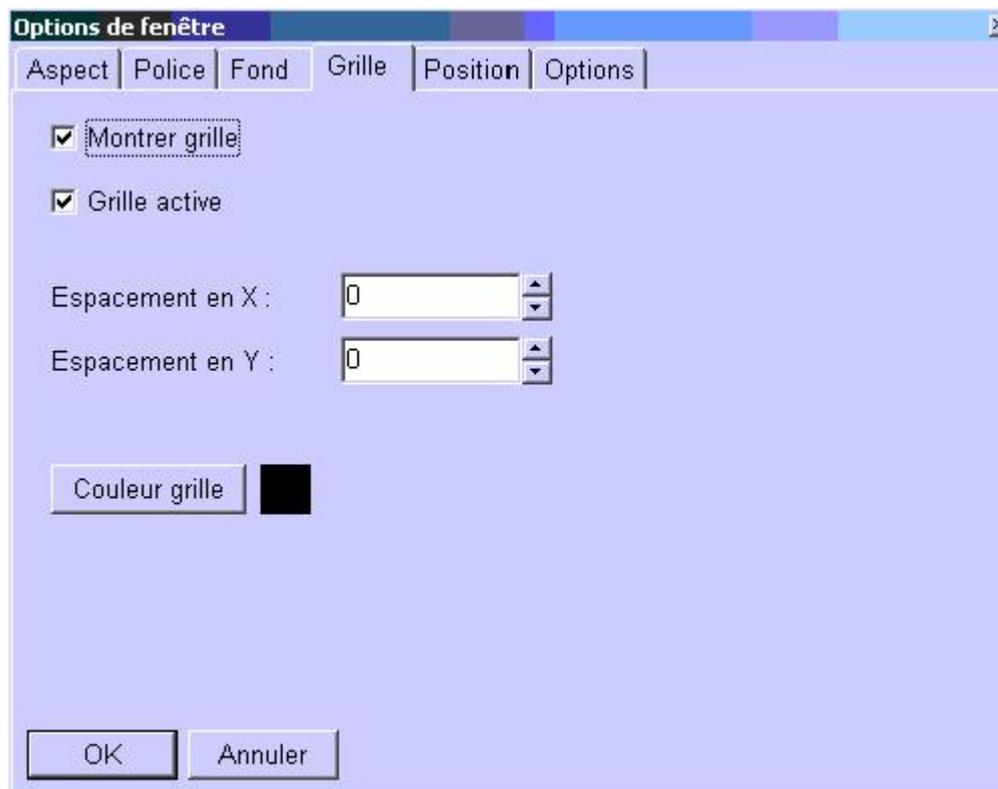
Delete the supervisory form background picture if one was selected.

Picture position

The user can select here the way that the picture must be displayed :

- Picture on left bottom
- Picture centered
- Mosaic : duplicate the picture in the whole supervisory form background
- Stretched picture : the picture is automatically stretched on the whole supervisory form background.

3.2.3.4 Grid



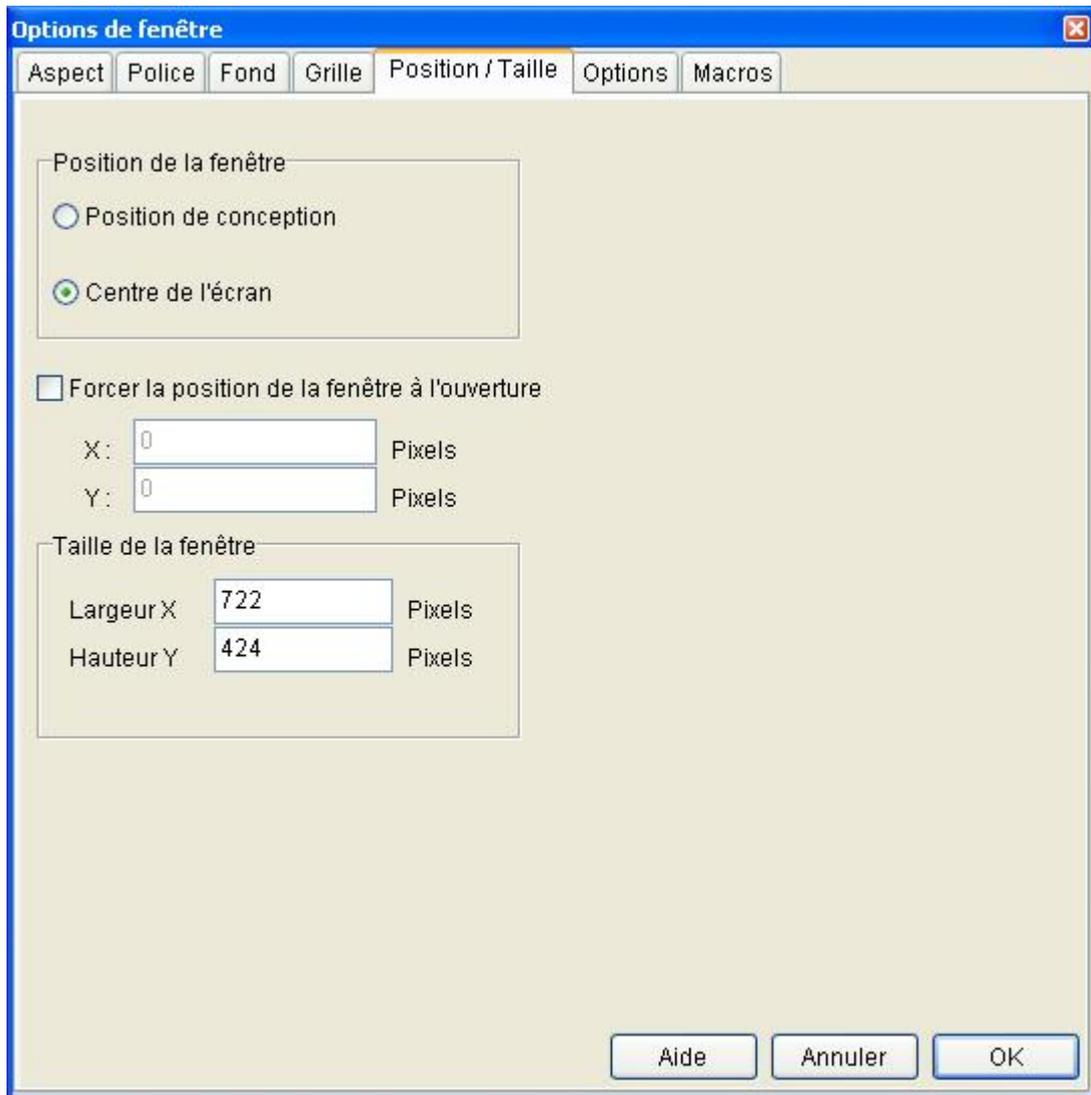
A grid can be displayed in designing mode. The grid is used to define a "resolution" to insert and to move the objects on the form. With the grid, it's easier to insert some objects to regular positions (for example, each 8 pixels)

The inserted objects are automatically inserted to the positions of the grid, their size is also adjusted to the resolution defined by the grid.

The user can modify the resolutions (X and Y), the grid is then repainted with the new resolutions. The grid will be displayed only if the X and the Y resolution are defined and at least equal to 2 and if the option "To show grid" is checked. In the same way, the grid is only active if the option "Active grid" is activated.

The grid is only displayed in designing mode.

3.2.3.5 Position / Size



This tab permits to choose the default position and size of the supervisory form when it is showed in execution mode.

Position

- Designing position (Position given to the form in the supervisory workshop)
- Center of the screen

Option "Specify supervisory form position at startup"

If this option is checked, the window will be displayed at the specified position in fields X and Y.

Size

The size (width and Height) of the window is displayed. The user can modify these values.

3.2.3.6 Options

This dialog box permits to define some options of the supervisory form available in execution mode.

Apigraf/OPTIMA PLC default IP address :

The user can define a default IP address of the OPTIMA PLC or APIGRAF PLC workstation in the supervisory form. If this field is empty, the project default IP address will be used. (Unless a default IP was defined in a supervisory object or in the expression)

It's also possible to insert a reference to a parameter in this field (example: {1} See Section " Open a supervisory form with parameters ")

Default OPC server name

Enter here if needed the name of the OPC server used by default to value the variables on the form. Caution : if an IP address is defined, the OPC server won't be taken in account.

Window initially maximized

The window will appear in full screen mode at startup.

Window initially minimized

The window will appear minimized mode at startup.

Scroll bars so necessary

This option defines if the content of the window can be scrolled by the user. For it, some scroll bars appear below and on the right of the window, what permits to displace the visible zone of the window.

These scroll bars are displayed only if the content of the form is not entirely visible.

Modal window

If this option is checked, at the opening, the window will be executed in modal mode. The other windows already

opened in the supervisory will stay at background and the modal window will stay on top until the modal window closing.

If another supervisory form is opened from a modal supervisory window, this other window must be a modal one.

Stay on top window

If this option is checked, the windows will always be displayed on top. All other windows which does not have this option will be displayed at the background. Even if the window is on top, the user can activate other window at the background.

Window may be :Maximized :

If this option is not checked, there will be no maximize button on the window. This window will not be able to be maximized.

Minimized :

If this option is not checked, there will be no minimize button on the window. This window will not be able to be minimized.

Closed :

If this option is checked, the Windows will not be able to be closes by the user.

Open several times :

If this option is checked, to every new call of the supervisory form (with SUPERG(FormName)), a new supervisory form will be open.

If this option is unchecked and the called window is already opened. The opened window will be bring to front of the screen and no other window will be created.

Window auto close :

If this option is checked and that the time specified in the correspondent field in seconds is then different from 0, the window will close itself automatically after n seconds according to the value entered in number of seconds after its opening.

This option can be useful for example to create a temporary message that is only displayed a few seconds.

Password number :

Defines a password number needed to allow the supervisory form opening.

If the field is empty or the value is 0, no password will be needed to open the supervisory form.

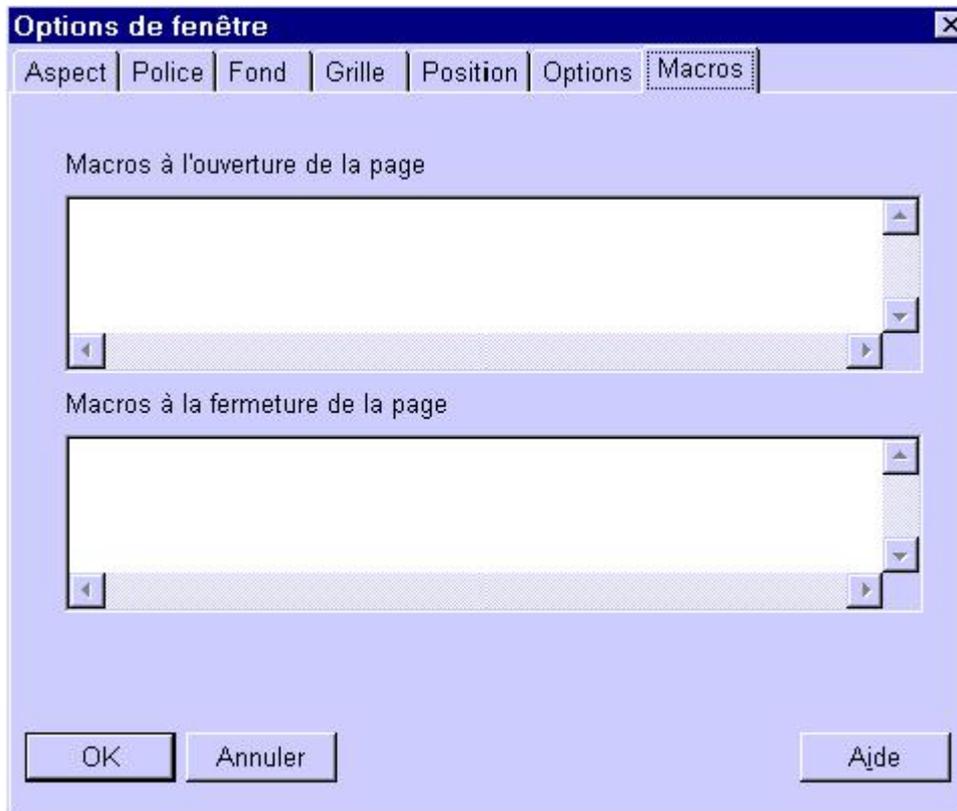
This password is defined using the local Macro : ChnMDP.

Min user level to open the form:

This option is only available and has only effect when the user management is enabled in Optima View (See [Generalities](#)^[306] for users management)

If a min level group is defined, only the group defined or upper in terms of rights will be able to open the supervisory form.

3.2.3.7 Macros



Opening window macros

Defines the macros that will be executed at the supervisory form opening.

caution : Enter in the field one macro by line!

It's also possible to insert a reference to a parameter in this field (example: {1} See Section " Open a supervisory form with parameters ")

" Macros to the closing of the page "

Input Box

Defines the macros that will be executed at the supervisory form closing.

caution : Enter in the field one macro by line!

It's also possible to insert a reference to a parameter in this field (example: {1} See Section " Open a supervisory form with parameters ")

To see the available local macros, see [Local macros](#) ^[29]

3.2.4 Supervisory workshop tool bars

In designing mode, the workshop contains some tool bars :

Object Selection ToolBar :



This tool bar is used to select the next object to insert on the supervisory form in designing mode. Select first the tab of the object family (For example : Standard) and select then a single object on the tab, the button of the object is then down confirming that the object has been selected.

The user make a left click on the supervisory form at the insert position to put the new created object. The object button on the toolbar is then released.

Function panel :



Button #1 : Shows the main option window of the current supervisory form. This option window can also be opened by making a right click on the supervisory form.

Button #2 : Object zoom function : This function is used to modify the object size with a zoom factor (entered as a percent value : exemple 200 to multiply the size and 50 to divide the size by 2.

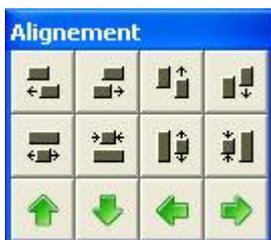
Button #3 : Set the current supervisory form in execution mode. In this case the button stay down, press a second time the button to put the current supervisory form in designing mode.

Button #4 : Set the current supervisory form in continuous update mode : In this mode, the supervisory is in designing mode but some objects when possible (Example : Input Boxes) display the refreshed value of the PLC variable or expression.

Button #5 : Delete all selected objects.

The function panel also displays the coordinates of the mouse pointer on the current supervisory form.

Alignment Panel:



The alignment tool panel is used to align some objects. Some functions are available if more than 2 objects are selected on the current supervisory form. Some other (the arrows) are enabled only if at least one object is selected on the supervisory form.

The two first rows are used to align several object depending on the min max positions or dimensions of the selected objects.

Example : the first button put all object at the same X position of the object with the X min value.

In this case the arrows move the selected objet in the requested direction for 1 pixel at each time. These functions are used to place the objects accurately on the supervisory form.

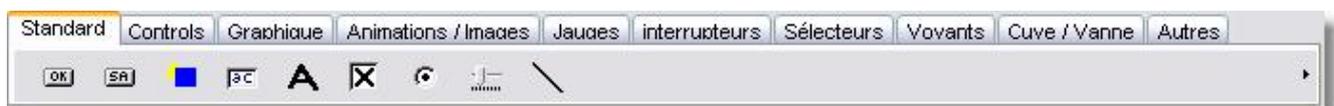
3.2.5 Supervisory objects management

The following manipulations are common to all objects of supervisory.

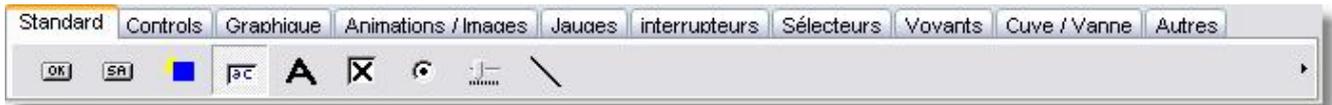
The user can create, modify, move, copy, paste, cut, resize, ... these different supervisory objects.

3.2.5.1 Create a new supervisory object

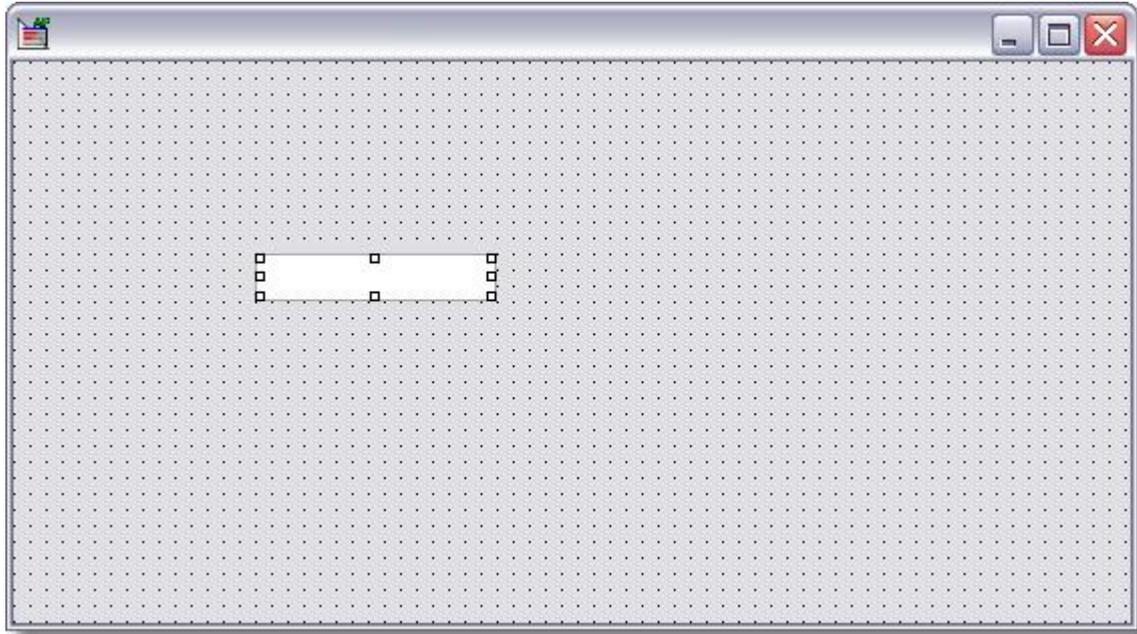
To create a new object, use the object tool bar in the main menu :



Select the tab that contain the object to create and then click on the object to create. The button of the selected object stay down confirming that the object has been correctly selected :



Move the mouse pointer at the position desired to insert the object and click on the left button. The new object is created at the pointed position. The object button on the object tool bar is automatically released.



3.2.5.2 Select a single supervisory object or a group of supervisory objects

To be modify, a supervisory object must be selected first.

When an object is selected, some drag rectangles appears on his corner.

When a single object is selected, 8 drags are created on the object, the user can use the drags to resize the object with the mouse.

Several objects can be selected simultaneously. In this case, the selected object will displayed 4 rectangles on their corners.

Selection by designation

A mouse left button click on one object select a single object. Another mouse left button click on a single object already selected unselect the object.

If the left button click is done with the SHIFT key keep pressed, the object is added to the previous selection.

If the left button click is done with the SHIFT key keep pressed on an already selected object, the object is removed from the previous selection.

Selection by including rectangle

It is possible to select in only one operation all objects situated in an rectangle zone.

For it, to click with the mouse on one of the corners of this zone (it must not be on an object), then keeping the mouse button pressed, move the mouse pointer on the other corner.

The objects situated in this defined rectangle are then selected. The objects partially contained by this rectangle are also selected.

3.2.5.3 Moving and resizing a supervisory object

Moving

Before being moved, an object must be selected.

To displace a supervisory object, select it by clicking on it, then to click again and without releasing the button of the mouse, move the object at the desired position.

If a grid of resolution is defined, the object can be moved only at positions defined by the grid.
If several objects are selected, they are moved simultaneously.

Resizing

To be resized, a supervisory object must be selected.

To resize the object, use the drags on the object's corners to resize it with the mouse pointer.
Put the mouse pointer cursor on an object drag (side or corner), click on the drag keeping the mouse button pressed and then move the cursor pointer to the selected position.

Direct input of the position and size

In some cases, the position and the size of an object must be entered numerically with the keyboard to get a more precise result.

For it, make a right click on the object to display the popup menu.

In the popup menu, choose " Position ". A dialog box is displayed to enter the position and dimensions numerical values (in pixels) : PositionX, PositionY, Width, Height.

Variable Position and size of the object

It's possible to define the positions and the sizes of the objects with PLC variables. In this case, the positions and size of such objects will be valued with the associated (Registers) variables.

Enter some variables or expression (R30 for example) in the position and dimension fields:
PositionX, PositionY, Width, Height.

The position and dimension of the object will be automatically updated with the variable values.

3.2.5.4 Resizing objects with the zoom function

To resize one or several objects, first select the objects to resize.

On the function panel,click on the button " Zoom ". The following window is displayed :

Dialog Box : "Change Scale"



The fields are the next one :

Zoom value

Enter here the zoom value as a percent value. The default value entered is 100 (for 100% - no resizing)

Option " Change object text size "

If this option is checked, the text font will also be resized. The font size of the text will be modified.

3.2.5.5 Delete supervisory objects

Select first the objects to delete.

Then click on the "Delete" button on the "Fonctions" panel to delete all selected supervisory objects.

It's also possible to press the key " Suppr " on the keyboard, that has the same effect that the click on the trash

button.



3.2.5.6 Bring to front or Send to back an object

Bring to front an object

Make a right click on the object to open the popup menu. Then select "Bring to front" in the popup menu. The object will be brought to front.

There are two categories of object, graphical controls and win controls. Win controls are always on top (example input boxes), and in some cases the "Bring to front" command has no effect.



Send to back an object

Make a right click on the object to open the popup menu. Then select "Send to back" in the popup menu. The object will be sent to back.

There are two categories of object, graphical controls and win controls. Win controls are always on top (example input boxes), and in some cases the "Send to back" command has no effect.



Caution : Some control objects (such as input boxes) stay always on top.

3.2.5.7 Duplicate objects with the mouse

It's possible to duplicate one or several objects with the mouse.

First select the objects to duplicate. Then keep pressed the "CONTROL" key on the keyboard. Click and keep down the left button of the mouse on one of the selected objects. Move the mouse pointer to another position on the supervisory form. Then release the mouse left button at the desired position. The selected objects are duplicated at the pointed position.

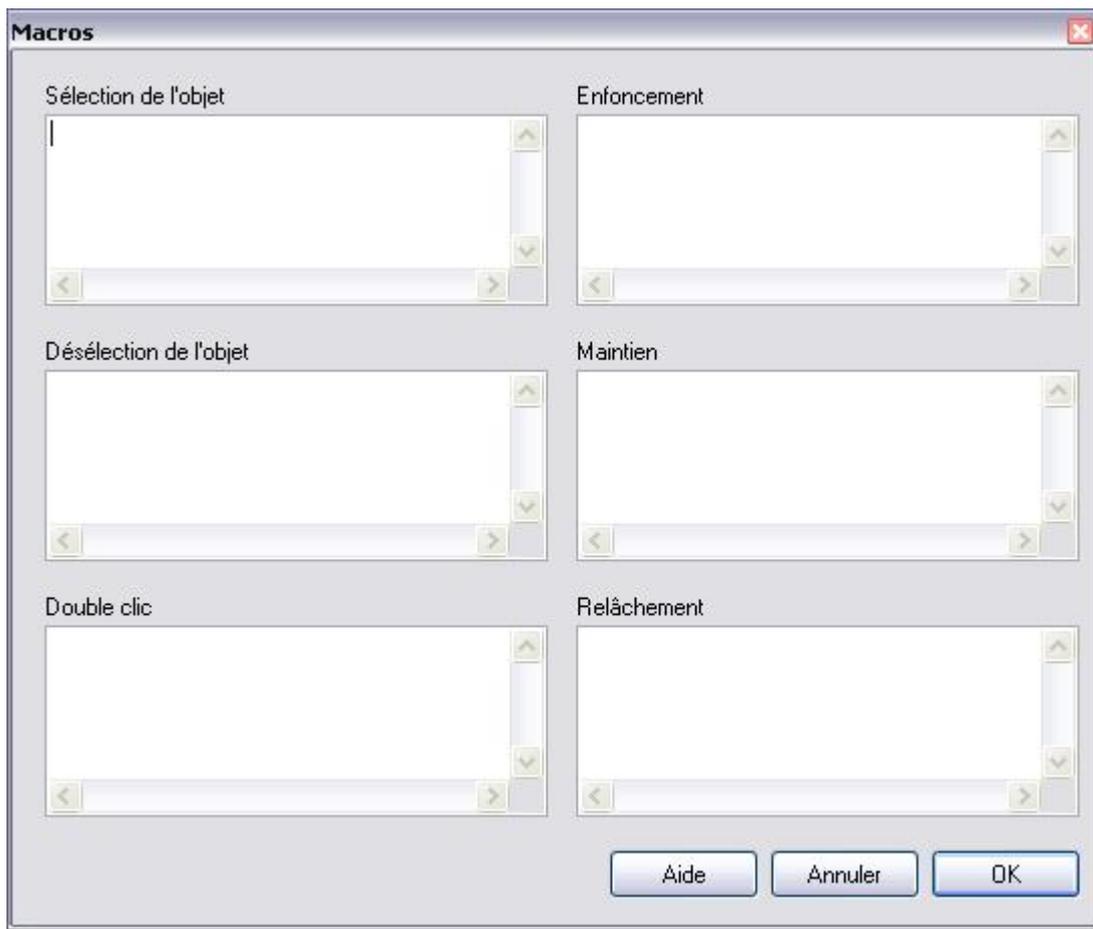
3.2.5.8 Define macros in objects events

The macros are some local or distant procedure that are executed with some object events.

Example : For a button, when the button is released, the release macros defined in the object are executed.

To define the object events macros :

Make a right click on the object to display the popup menu. Select "Macros" in the popup menu.
The macro window is displayed :



The defined macros can be executed with several object events :

Object Focus

Object UnFocus

Objet double click

Object Mouse down

Object Mouse up

For more details on the available macros, refer to the corresponding chapters.

3.2.6 Usual objects for supervisory form designing

3.2.6.1 STANDARD tab objects

3.2.6.1.1 Text

This type of object is used to display a static text.

The text can also be dynamic if an expression or a variable is defined in the object.

Dialog box "Simple Text"



The different parameters are :

Text

Enter here the static text

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

"Automatic size" Option

When this option is checked the size of the object is automatically adjust to the size of the text.

"Automatic word wrap" option

If this option is checked, the text will automatically be written on several lines if the whole text doesn't hold on only one line depending on the object width.

Font

This button open the font dialog box. The user choose here the font used for displaying the text.

Back color

Open the color dialog box. The user defines here the background color of the text object.

Transparent "Option"

If this option is checked, the text will appear without background. If the user put a transparent text on a picture, the text will be " melted " in the picture. In this case the background color parameter has no effect.

Horizontal justification

Permits to choose the horizontal alignment (left, right or centered)

Vertical alignment

Permits to choose the vertical alignment (left, right or centered)

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Option " Dynamic text "

If this option is checked a memo is displayed to input the expression which will be used to refresh the text.

The text will then be up to dated dynamically with the value of the expression.

" Expression "

Enter here the expression that can be of any type (Chain, register, flag, etc.).

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Neutralization expression

The user can specify the variable or the expression of neutralization (This one must be a Boolean one). The text object will not be visible when the variable or the expression value will be set to TRUE.

Example : SA10

It's possible to specify the distant IP address of the APIGRAF/OPTIMA PLC workstation in the expression. In this case, this address will be taken in account to value the expression.

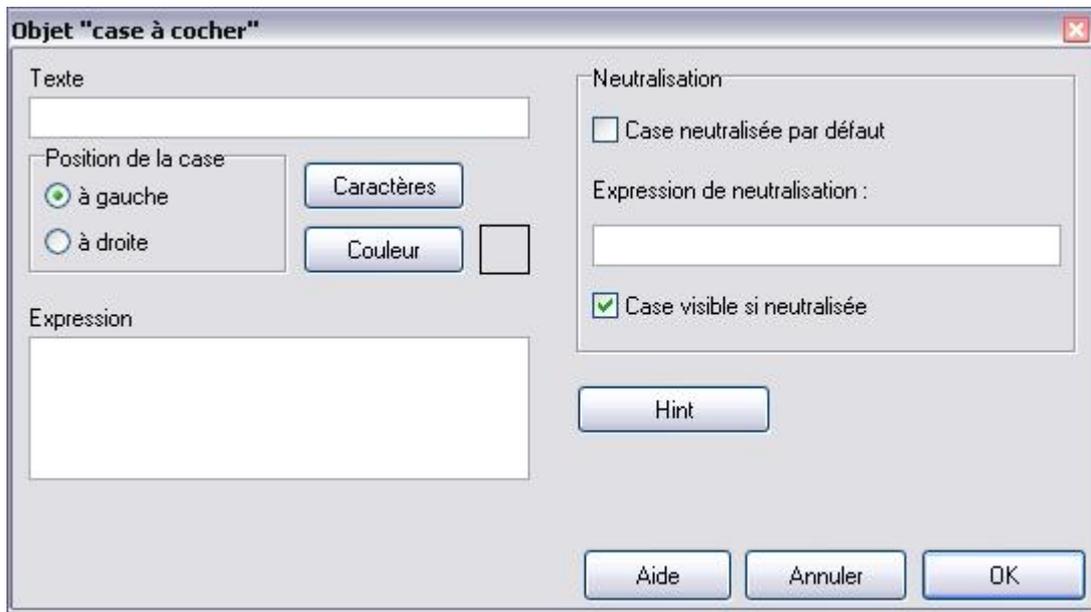
Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.1.2 Check box

The object is a check box used to control a PLC variable or expression which type is boolean.

Dialog box "Check Box"



The parameters are :

Text

Defines the text of the check box.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Position

Indicate the position of the text in relation to check box (on the left or on the right).

Font

Defines the font of the text displayed in the object "Check box". The button displays the font dialog box.

Color

Select here the check box background color.

Expression

Enter her the PLC Variable or expression to link to the check box. The type of the variable must be Boolean.

Example : L10

It is possible to specify the address distant IP of the kernel APIGRAF PLC in the expression. If it is the case, it is this address that will be taken in account for the cooling.

Example : !193.0.1.1 !L10

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Box neutralized by default

If this option is checked, it won't be possible to modify the state of the checkbox by clicking over. The box will only permit to display the state of the variable associated. Besides the checked box will appear grey.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The checkbox will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the distant IP address of the APIGRAF/OPTIMA PLC workstation in the expression. In this case, this address will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is checked, the check box will be visible when it will be neutralized. However, it won't be enabled and the user won't be able to control it. (The check box is grayed)

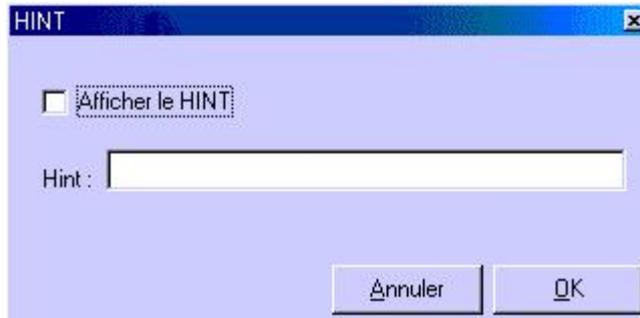
If this option is not checked, the check box will not be visible when neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.1.3 Radio group

The Radio group is a list of option in which only one option can be selected at tone time. A PLC variable or expression can be controlled by this object. (The variable type must be numerical and Integer)

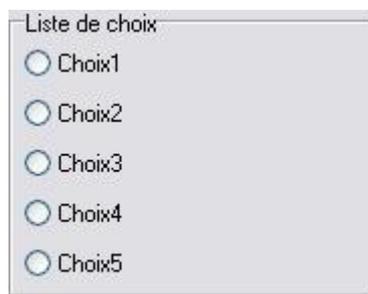
Example : R12

It is possible to specify the address distant IP of the kernel APIGRAF PLC in the expression. If it is the case, it is this address that will be taken in account for the cooling.

Example : !193.0.1.1 !R12

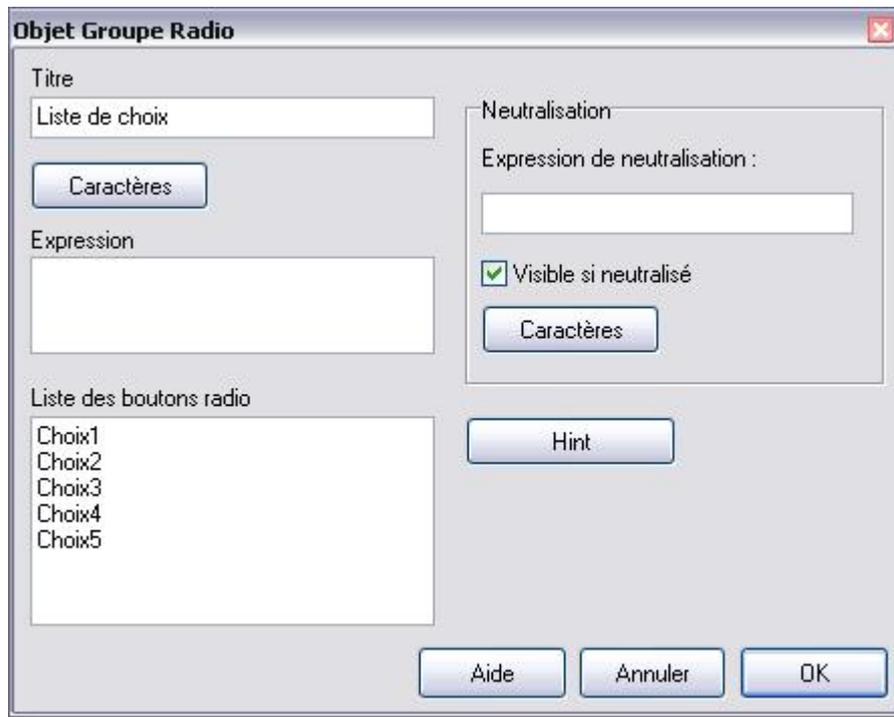
The register is affected by the index of the selected option (The index begins form 0 for the first option).

Example :



In this example, if the associated variable is R3, the value 0 is affected in R3 if the option #1 is selected. The value 1 is affected in R3 if the option #2 is selected

Dialog Box "Radio Group"



The parameters are:

Title

Enter here the title of the radio group. The title is displayed above the options list of the radio group.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Font

Select here the font used to display the texts of the radio group radio

Expression

To indicate in this field the variable PLC to use (of integer numeric type)

Example : R12

It is possible to specify the address distant IP of the kernel APIGRAF PLC in the expression. If it is the case, it is this address that will be taken in account for the cooling.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

One will note that if the list only includes two buttons radios, one will be able to use a variable PLC of boolean type since the possible values will be 0 or 1.

List of the buttons radio

To enter the designations of the buttons directly in their order of apparition. To validate an input with " ENTERED " to pass to the following line.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The group radio will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It is possible to specify the address distant IP of the kernel APIGRAF PLC in the expression. If it is the case, it is this address that will be taken in account for the cooling.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is active, the group radio will be visible when this one will have neutralized. It will be however impossible to manipulate the group radio.

If this option is not checked, the groups radio will be invisible when it will have neutralized.

Button " Characters " in the window " Neutralization "

Permits to define the font used to display the group radio when this one neutralized and that the option " Visible if neutralized " is checked.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

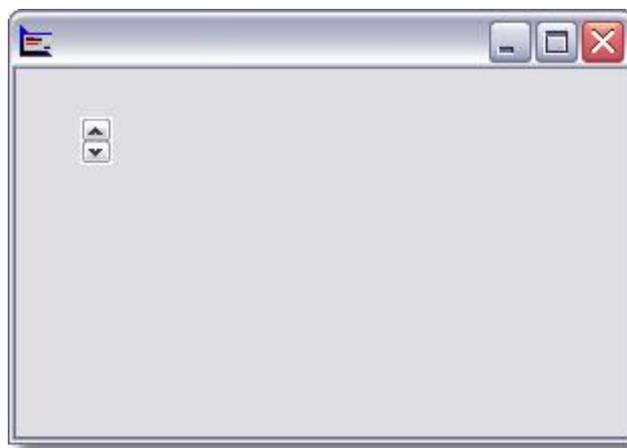
« Hint »

Enter here the "Hint" text.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.1.4 Up/Down arrows

This object can decrement/increment an integer variable by clicking on the Up and Down arrows.



Dialog box "UP/DOWN Arrows" :

The parameters are :

Expression

Enter here the PLC variable or expression to be controlled. This variable type must be Integer (Registers).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Min

This option defines the minimal value. It is the value below which one won't be able to descend by manipulating the arrow of the low.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

In this case, the boundary-mark min of the cursor will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Max

This option defines the maximal value. It is over the value to the of which one won't be able to descend by manipulating the arrow of the top.

This field offers the same options of use that the field " Value Min " here over.

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Value Increment

Specify the value of increment / decrement to every click on the arrows. The value is by default 1.

This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Orientation

This option permits to define if the arrows must be oriented horizontally or straight up.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The arrows will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is checked, the arrows will be visible when the object will have neutralized. However, the operator won't be able to act over.

If this option is not checked, the object will be invisible when it will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.2 BUTTONS tab objects

3.2.6.2.1 Simple button

This object is also called " Windows button " and is generally used to execute local macros as :

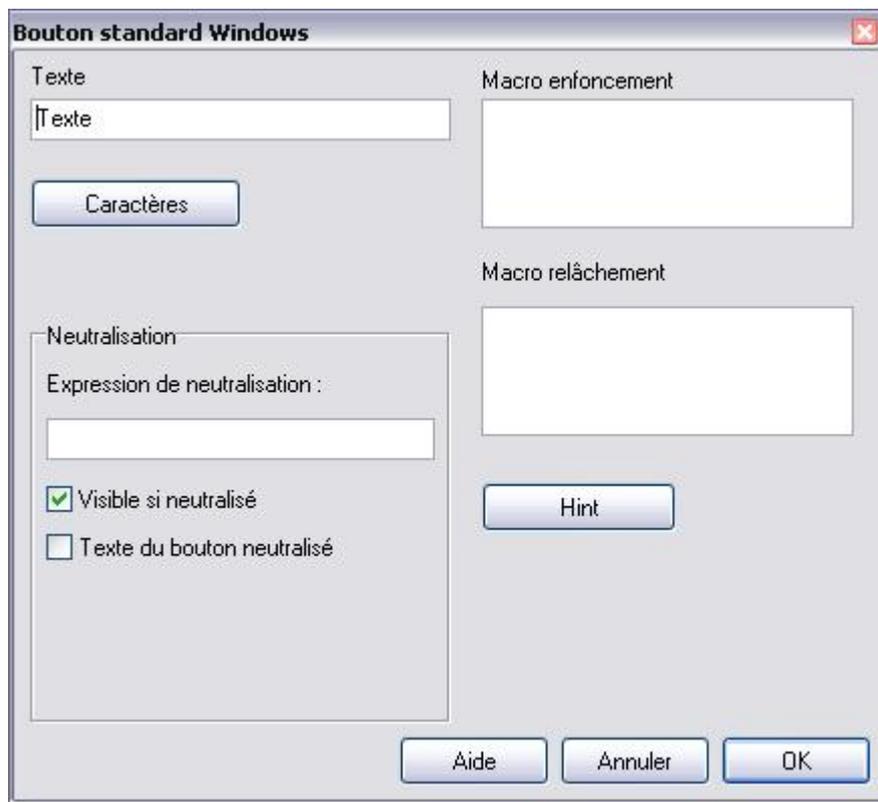
- To open another supervisory form
- To close the page, etc.

The macros can be executed on distant APIGRAF/OPTIMA PLC for distant macros.

Example : SETSA1:=1; (See the chapter "Macros" for more details).

The button can now control a PLC variable (Type Boolean). This button is only monostable.

Dialog box simple button



The parameters are :

Text

Enter here the text of the simple button

A keyboard shortcut (ALT + a letter) can be affected to the simple buttons. The user must add in the button text, the symbol "&" before the letter that will act as shortcut.

Example : if the button text is: M&odules: the shortcut of the button will be ALT + O

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Font

Defines the text font of the button. Opens the font dialog box.

On Press Macro

This control permits to enter the macros quickly to execute at the time of the forcing of the button.

On Relax Macro

This control permits to enter the macros quickly to execute at the time of the forcing of the button.

It's also possible to insert a reference to a parameter in the macros (example: {1} See Section " Open a supervisory form with parameters ")

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The button will have neutralized every time then that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is active, the button will be visible when it will have neutralized but this one won't be able to be manipulated. Besides the text of the button will appear in grey tints.

If this option is not active, the button will be invisible as soon as the expression of neutralization is to TRUTH (So of course, the expression of neutralization is defined).

Option " Text of the button neutralized "

This option must be checked if one wants to have a different text on the button when this one neutralized.

As soon as one nicks this option, an input window appears below this option, this window permits to enter the text of the button neutralized.

Macro laxity

This control permits to enter the macros quickly to execute at the time of the laxity of the button.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Expression (Optional)

Enter here the PLC variable or expression controlled by the button (the variable type must be Boolean)

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If a an expression is specified, the corresponding PLC variable will be set to 1 when the button will be activated with the mouse. The PLC variable will be reset when the button will be relaxed.

3.2.6.2.2 API button

This object is a button which can be monostable or bistable (can stay pressed after activation)

The API Button can control a PLC variable or expression. (of Boolean Type)

The button can also have different shapes (rectangle or circled)

Dialog Box : API Button
Dialog box - Button API: General

The parameters are :

Expression

Enter here the PLC variable or expression controlled by the button (the variable type must be Boolean)

In Execution mode or in refreshing designing mode, the API button is neutralized if the expression or the variable cannot be valued on the PLC.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Button type

Specify here the button type : monostable, bistable or neutralized. With the monostable mode, the button is released as the mouse button is released. With the bistable mode, the button keep pressed after one mouse click down and the button is released at the next mouse click down on it.

With the neutralized mode, the user won't be able to control the button. However the button will appear pressed or released according to the value of the expression controlled. In this case, the button properties (color etc.) are defined in the neutralization tab of the dialog box.

" Relaxed button " options

These options define the general display of the button in the relaxed state. The parameters are :

• Color

Defines the button back color when relaxed

• Text

Defines the text on the button when relaxed.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

• Font Button

Open the font dialog box. Defines the font of the text displayed on the button when relaxed.

• Picture Button

Defines a background picture for the button when relaxed. Opens the image file dialog box.

• " Automatic transparency of the picture " Option

If this option is checked, the background picture is displays with transparency. The transparency color is defined automatically.

• " Adjustment of the picture to the size of the button" Option

This option permits to adjust the size of the picture to the size of the button.

• " Delete Picture " Button

This control permits to remove the defined background picture.

" Pressed Button" Options

These options define the general display of the button in the pressed state. The parameters are :

• Color

Defines the button back color when pressed.

• Option " Text "

If this option is checked, the user can enter a different text to display on the button in the pressed state.

- **Text**

This control is only visible if the option " Text " is checked.

Defines the text on the button when pressed.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

- **"Font" Button**

This control is only visible if the option " Text " is checked.

Open the font dialog box. Defines the font of the text displayed on the button when relaxed.

- **" Picture" Button**

Defines a background picture for the button when relaxed. Opens the image file dialog box.

- **" Automatic transparency of the picture" Option**

If this option is checked, the background picture is displays with transparency. The transparency color is defined automatically.

- **" Adjustment of the picture to the size of the button " Option**

This option permits to adjust the size of the picture to the size of the button.

keyboard ShortCut

The user can specify a key on the keyboard which activate the API button. The valid shortcut keys values are :

Letters from A to Z, No case matching.

The numbers of 0 to 9

Functions KEYS of F2 (The F1 key being reserved for aide Windows®) to F12
INSERT, SUPPR, ESCAPE, BACK

The keys combinations with ALT: ALT TO, ALT B, ALT C,..., ALT Z, ALT 0. ALT 9

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



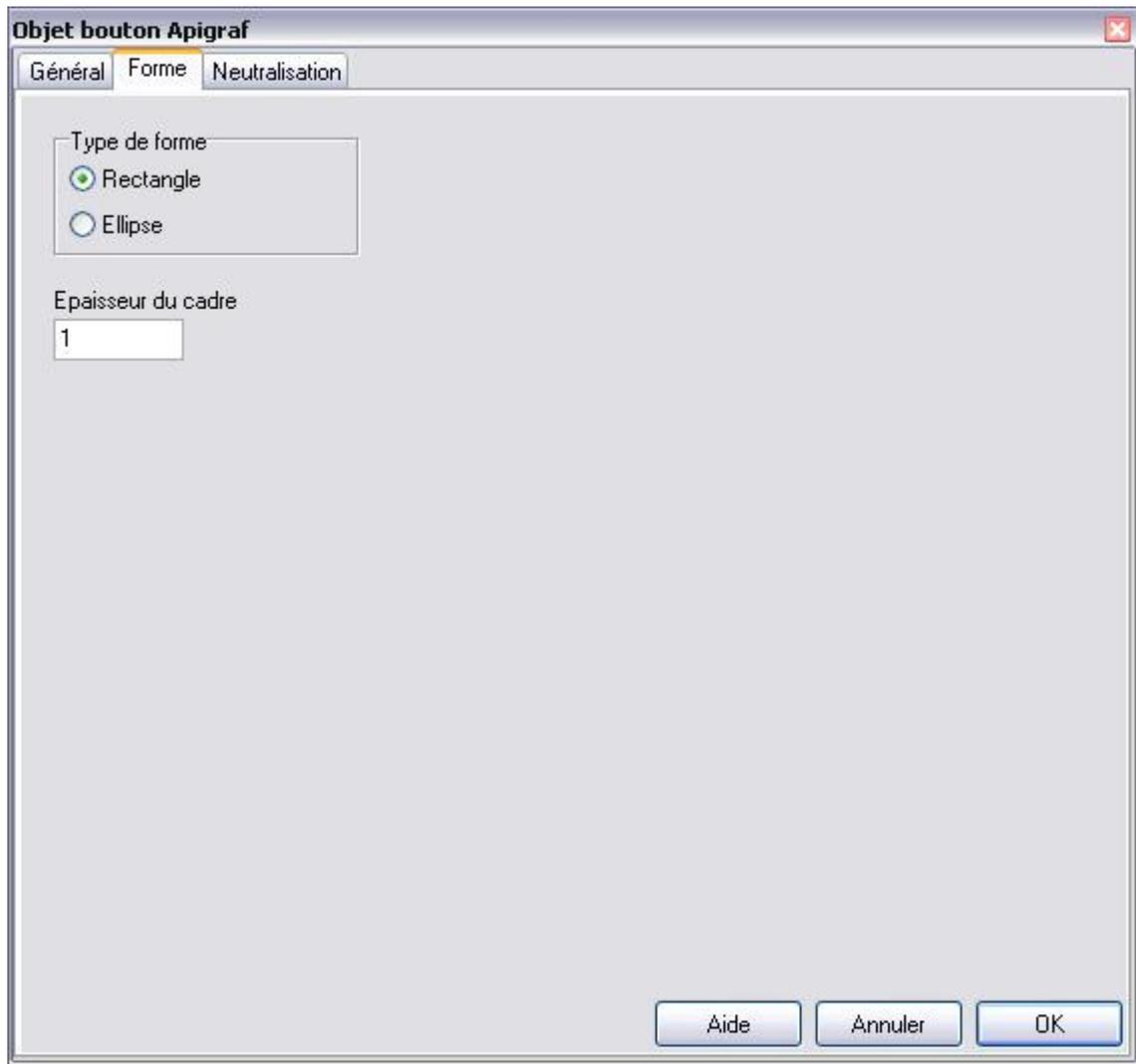
The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

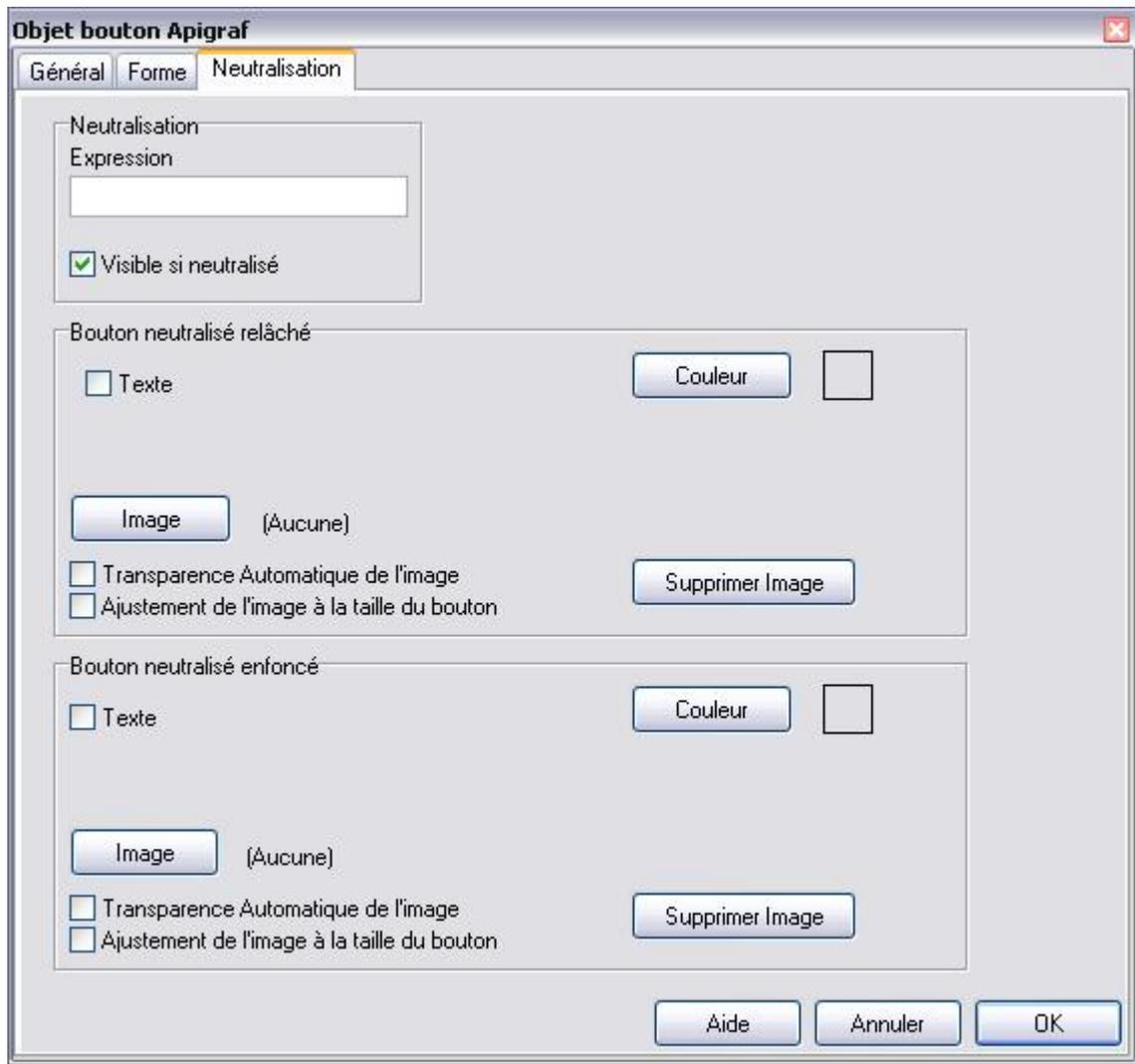
Dialog box : button API - Shape

The parameters are :

- **Shape**
Permits to choose the shape of the button between Rectangle or Ellipse.
- **Frame Thickness**
Define the thickness of the button frame. The default value is 1 pixel.

Dialog box : button API - Neutralization

This box is used to define the neutralization options of the API button.



The parameters are :

- **Neutralization expression**

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The button will be neutralized when the PLC variable will be set to True.

Example : L9

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !L9

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

- **"Visible if neutralized" Option**

If this option is not checked, the button will not be visible when neutralized.

- **" Neutralized relaxed Button" Options**

These options define the general display of the button in the neutralized relaxed state.

These options relates the color, the background picture and the text of the button when this one is neutralized and relaxed (The value of the expression of the button is then equal to 0)

- **"Neutralized pressed Button" Options**

These options define the general display of the button in the neutralized pressed state.

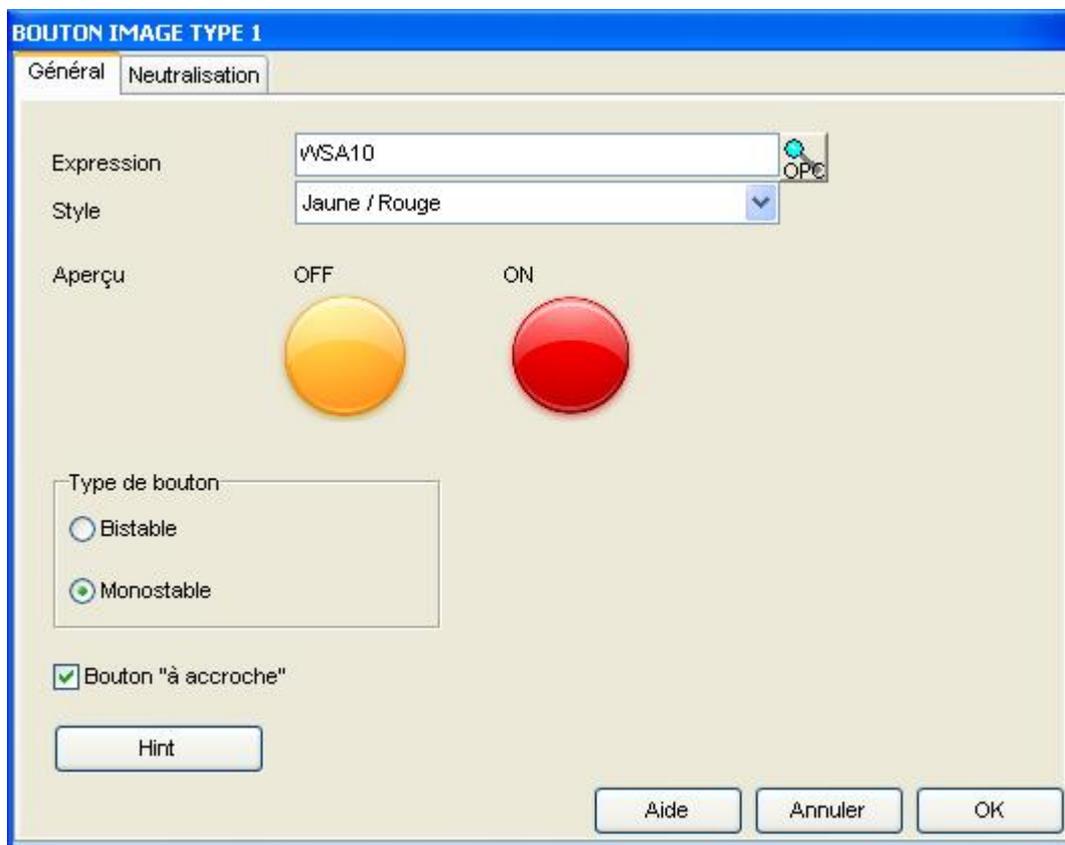
These options relates the color, the background picture and the text of the button when this one is neutralized and pressed (The value of the expression of the button is then equal to 1)

3.2.6.2.3 Push button type #1

This object is a rounded button used to control (set or reset) a variable which type is Boolean :



Dialog Box "Push Button type #1" - General:



The parameters are :

Expression

Enter here the PLC variable or expression controlled by the button (the variable type must be Boolean)

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Style

Permits to define the button color and shape : Blue, green, red or yellow, circle or rectangle.

Button type

Specify here the button type : monostable, bistable or neutralized. With the monostable mode, the button is released as the mouse button is released. With the bistable mode, the button keep pressed after one mouse click down and the button is released at the next mouse click down on it.

"Button to hangs" Option

If this option is checked, with the first click on the button, the PLC variable will set to True. The user won't be able to reset the button by clicking on it. The button will only be relaxed when the PLC variable will be reset to FALSE in the PLC

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

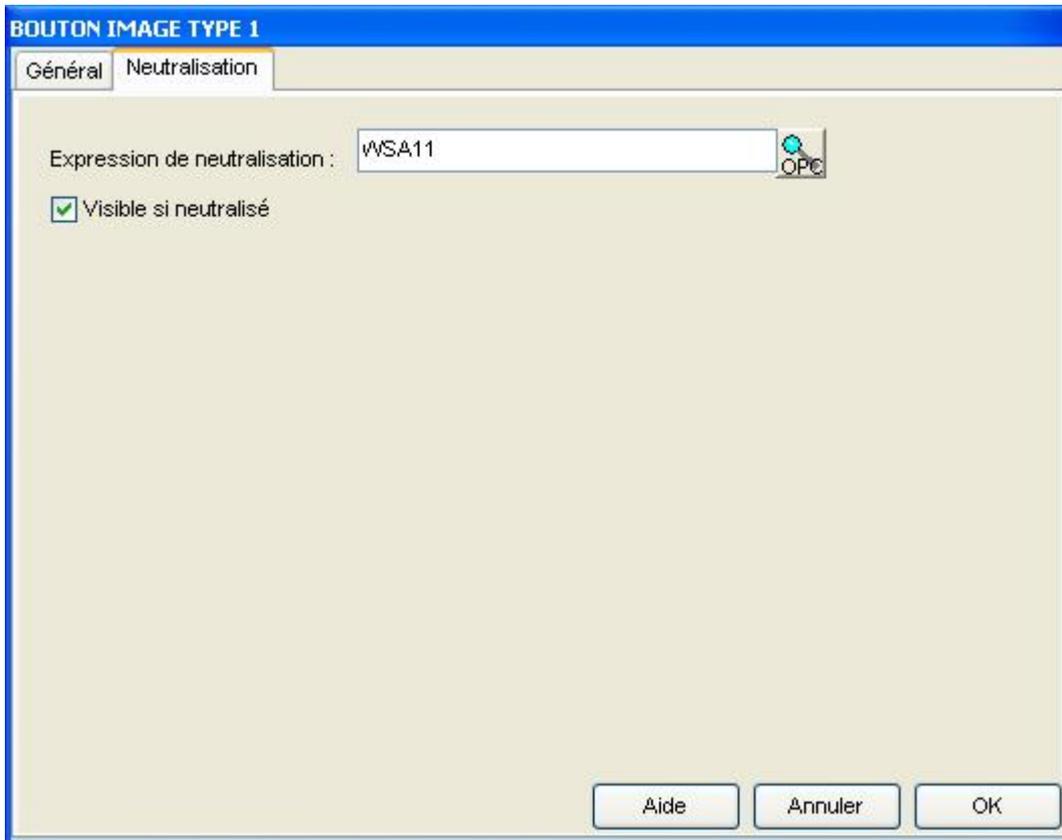
« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Dialog Box "Push Button type #1" - Neutralization:



Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The button will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the button will not be visible when neutralized.

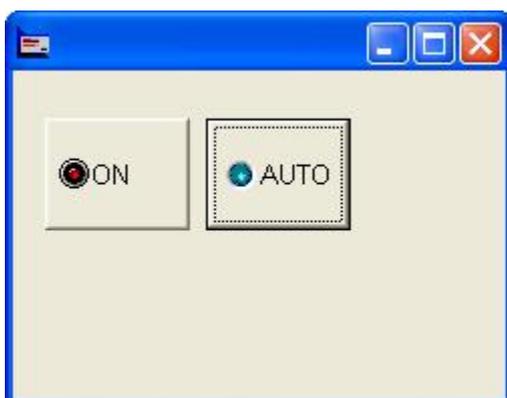
If this option is checked, the object will be visible but the user won't be able to control it when neutralized.

3.2.6.2.4 DEL button

This object is a standard button with a DEL that indicates the state of the boolean PLC variable or expression associated to the button.

The button can be monostable, bistable or neutralized by default.

A neutralization variable can also be defined.



Dialog Box "DEL Button" - General :
Expression

Enter here the PLC variable or expression controlled by the button (the variable type must be Boolean)

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Button Type

Specify here the button type : monostable, bistable or neutralized. With the monostable mode, the button is released as the mouse button is released. With the bistable mode, the button keep pressed after one mouse click down and the button is released at the next mouse click down on it.

With the neutralized mode, the user won't be able to control the button. However the button will appear pressed or released according to the value of the expression controlled. In this case, the button properties (color etc.) are defined in the neutralization tab of the dialog box.

Type of button

Defines the DEL (Blue, red, triangular,...)

" Font " Button

Open the font dialog box. Select here the font of the text on the button.

Keyboard Shortcut

The user can specify a key on the keyboard which activate the DEL button. The valid shortcut keys values are :

Letters from A to Z, No case matching.

The numbers of 0 to 9

Functions KEYS of F2 (The F1 key being reserved for aide Windows®) to F12
INSERT, SUPPR, ESCAPE, BACK

The keys combinations with ALT: ALT TO, ALT B, ALT C,..., ALT Z, ALT O. ALT 9

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

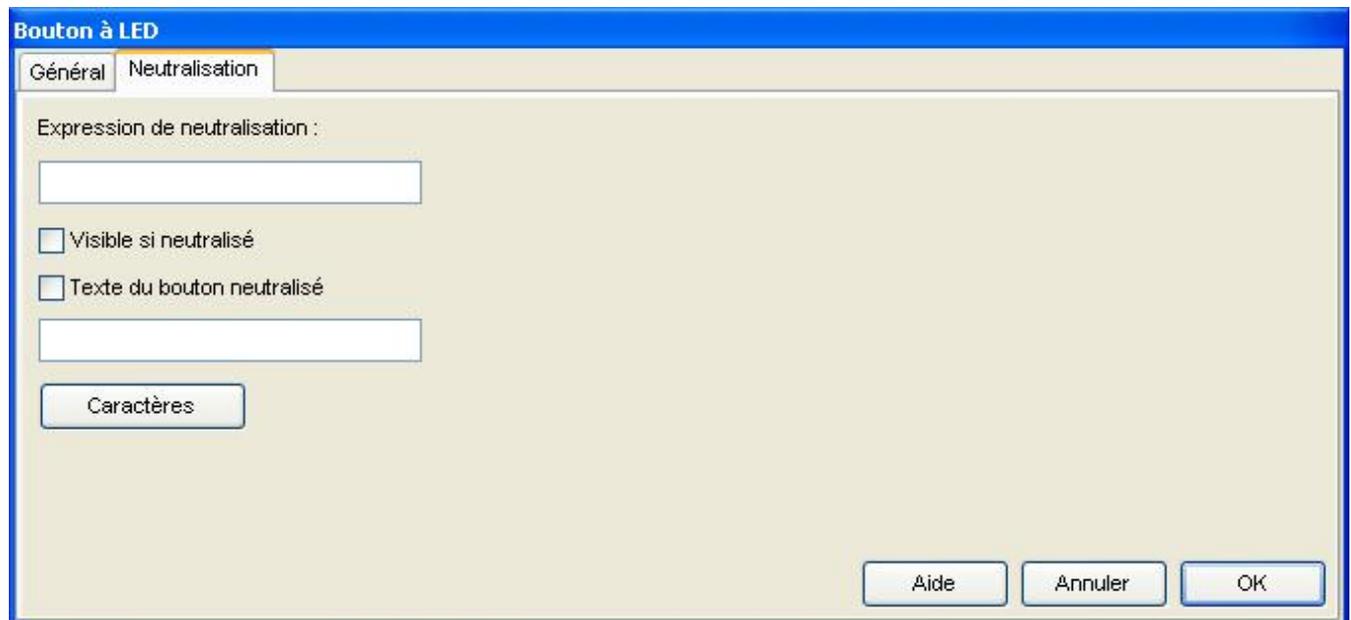
« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Dialog box "LED Button" - Neutralization :



Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The button will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is not checked, the button will not be visible when neutralized.

If this option is checked, the object will be visible but the user won't be able to control it when neutralized.

" Neutralized button text" Option

This option must be checked if the text of the button in the neutralized state must be different.

When this option is checked, an input box is enabled to enter the neutralized text. A font button is also enabled to select the font of the button text when neutralized.

3.2.6.2.5 Function button

This object is typically used to execute local macros :

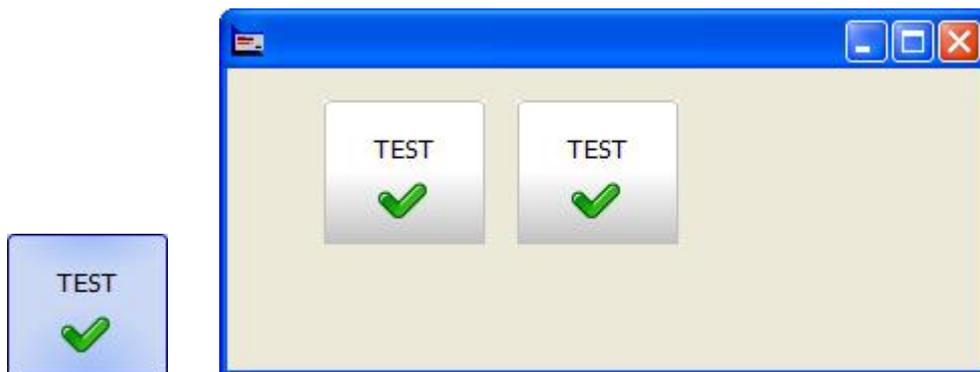
- To open another supervisory form : SUPERG (FormName)
- To close the current supervisory form : CLOSE , etc.

The macros can be executed on distant APIGRAF/OPTIMA PLC for distant macros.

Example : SETSA1:=1; (See the chapter "Macros" for more details).

The button can now control a PLC variable (Type Boolean). This button is only monostable.

This button can integrate a picture (icon, bitmap, PNG,...).



Dialog box "Function button" - General :

Text

Enter here the text of the function button

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Font Button

Defines the text font of the button. Opens the font dialog box.

On Press Macro

This control permits to enter the macros quickly to execute at the time of the forcing of the button.

On Relax Macro

This control permits to enter the macros quickly to execute at the time of the forcing of the button.

It's also possible to insert a reference to a parameter in the macros (example: {1} See Section " Open a supervisory form with parameters ")

Keyboard shortcut

The user can specify a key on the keyboard which activate the function button. The valid shortcut keys values are :

Letters from A to Z, No case matching.

The numbers of 0 to 9

Functions KEYS of F2 (The F1 key being reserved for aide Windows®) to F12
INSERT, SUPPR, ESCAPE, BACK

The keys combinations with ALT: ALT TO, ALT B, ALT C,..., ALT Z, ALT 0. ALT 9

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Expression (Optional)

Enter here the PLC variable or expression controlled by the button (the variable type must be Boolean)

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If a an expression is specified, the corresponding PLC variable will be set to 1 when the button will be activated with the mouse. The PLC variable will be reset when the button will be relaxed.

Dialog box "Function button" - Shape :**"Rounded button" option**

If this option is checked, the corners of the buttons will be rounded.

BOUTON DE FONCTION

Général **Forme** Neutralisation Icônes

Bouton Arrondi

Bouton au repos	Bouton sélectionné
Couleur départ fond gradient haut <input type="text"/>	Couleur départ fond gradient haut <input type="text"/>
Couleur fin fond gradient haut <input type="text"/>	Couleur fin fond gradient haut <input type="text"/>
Couleur départ fond gradient bas <input type="text"/>	Couleur départ fond gradient bas <input type="text"/>
Couleur fin fond gradient bas <input type="text"/>	Couleur fin fond gradient bas <input type="text"/>

Bouton enfoncé	Bouton enfoncé
Couleur départ fond gradient haut <input type="text"/>	Couleur départ fond gradient haut <input type="text"/>
Couleur fin fond gradient haut <input type="text"/>	Couleur fin fond gradient haut <input type="text"/>
Couleur départ fond gradient bas <input type="text"/>	Couleur départ fond gradient bas <input type="text"/>
Couleur fin fond gradient bas <input type="text"/>	Couleur fin fond gradient bas <input type="text"/>

Aide Annuler OK

It's also possible to define all colors used for displaying the function in its different states.

Dialog Box " Function button" - Neutralization :

Neutralization expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The button will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

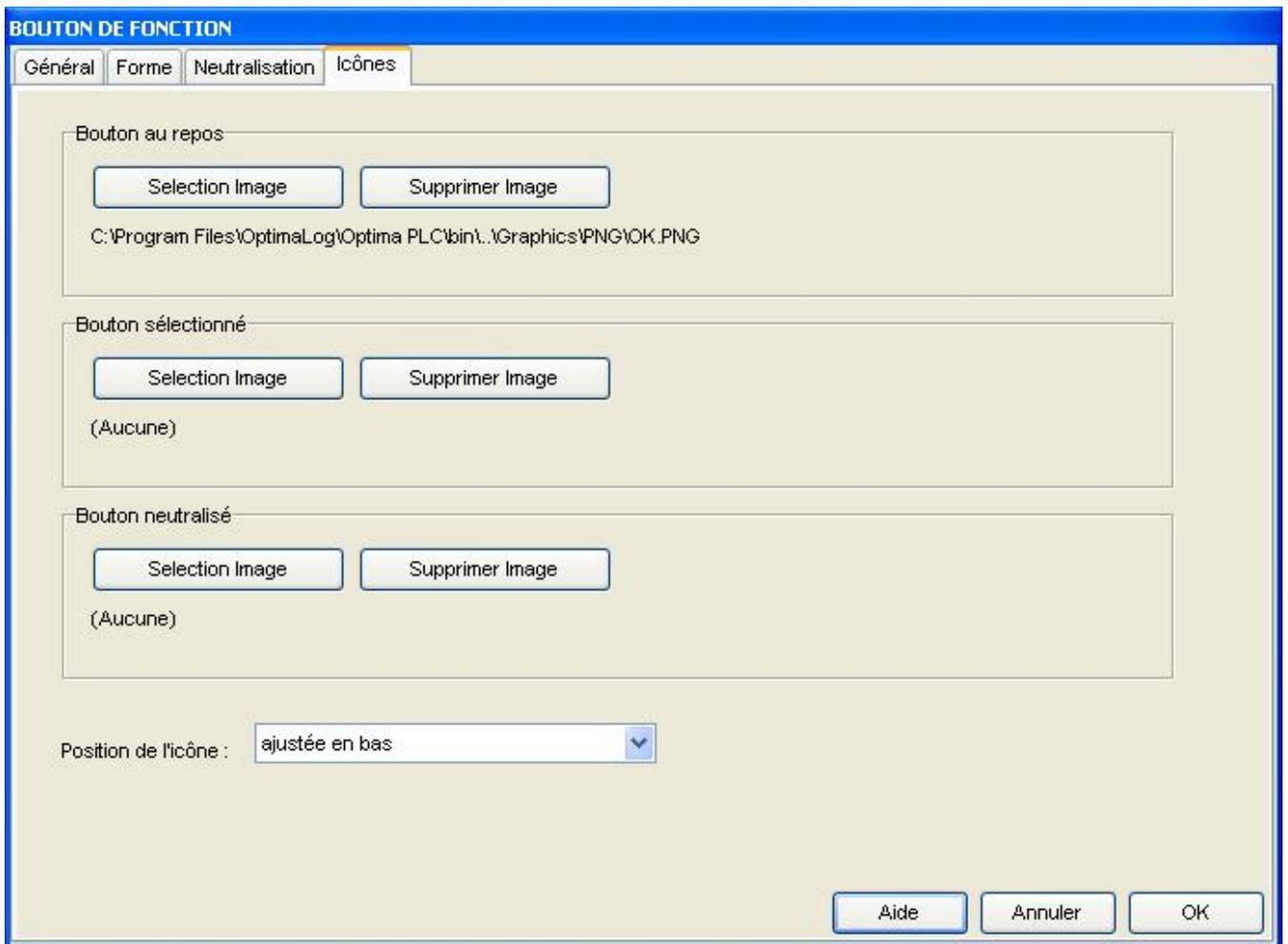
It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is not checked, the button will not be visible when neutralized.

If this option is checked, the button will be visible but the user won't be able to control it when neutralized.

Dialog Box "Function Button" - Icons :



Select here all the pictures to display on the function button. All formats are available, the PNG format is recommended. At the first picture selection, the open directory is by default the PNG Directory that contains a library of pictures in the PNG format.

It's also possible to specify the picture position on the button in the dropped list "Icon Position"

3.2.6.2.6 Translucent Function button

This object is typically used to execute local macros :

- To open another supervisory form : SUPERG (FormName)
- To close the current supervisory form : CLOSE , etc.

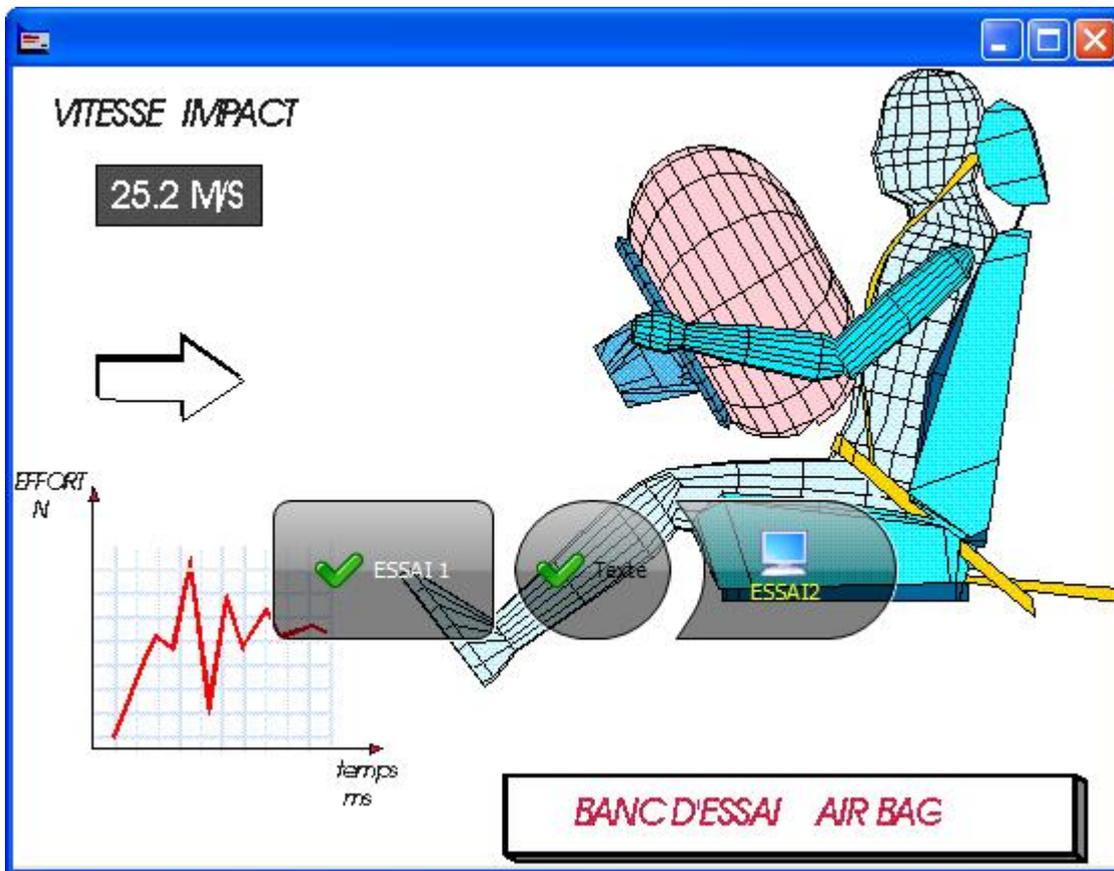
The macros can be executed on distant APIGRAF/OPTIMA PLC for distant macros.

Example : SETSA1:=1; (See the chapter "Macros" for more details).

The button can now control a PLC variable (Type Boolean). This button is only monostable.

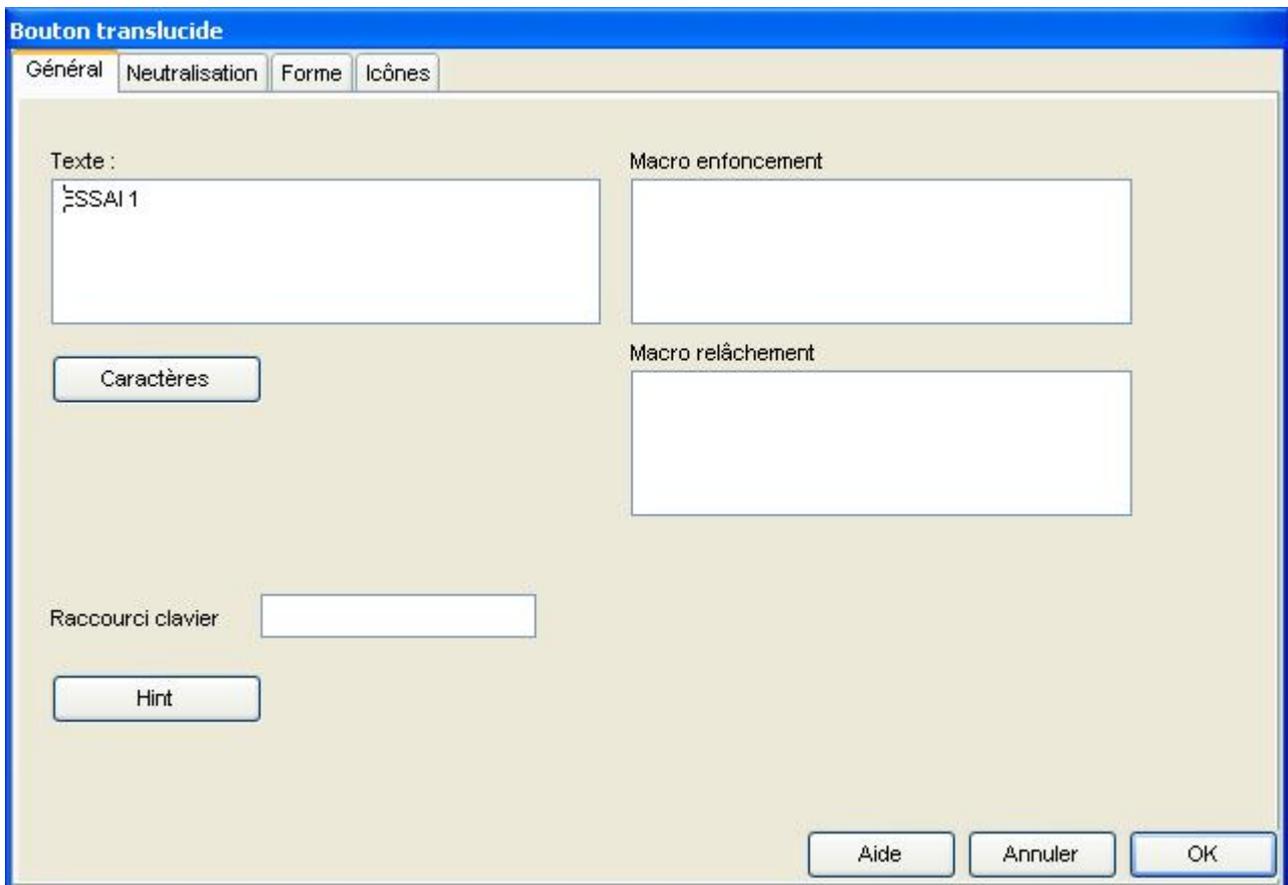
This button can integrate a picture (icon, bitmap, PNG,...).

It's appearance is translucent :



In this example, the buttons let appear the background picture of the supervisory form.

Dialog box "Translucent Function Button" - General :



Text

Enter here the text of the function button

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Font

Defines the text font of the button. Opens the font dialog box.

On Press Macro

This control permits to enter the macros quickly to execute at the time of the forcing of the button.

On Relax Macro

This control permits to enter the macros quickly to execute at the time of the forcing of the button.

It's also possible to insert a reference to a parameter in the macros (example: {1} See Section " Open a supervisory form with parameters ")

Keyboard shortcut

The user can specify a key on the keyboard which activate the function button. The valid shortcut keys values are :

Letters from A to Z, No case matching.

The numbers of 0 to 9

Functions KEYS of F2 (The F1 key being reserved for aide Windows®) to F12
INSERT, SUPPR, ESCAPE, BACK

The keys combinations with ALT: ALT TO, ALT B, ALT C,..., ALT Z, ALT 0. ALT 9

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Expression (Optional)

Enter here the PLC variable or expression controlled by the button (the variable type must be Boolean)

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If a an expression is specified, the corresponding PLC variable will be set to 1 when the button will be activated with the mouse. The PLC variable will be reset when the button will be relaxed.

Dialog box "Translucent Function Button" - Neutralization :

Neutralization expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The button will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is not checked, the button will not be visible when neutralized.

If this option is checked, the button will be visible but the user won't be able to control it when neutralized.

Dialog box "Translucent Function Button" - Shape :

The screenshot shows a dialog box titled "Bouton translucide" with four tabs: "Général", "Neutralisation", "Forme", and "Icônes". The "Forme" tab is selected. The dialog contains the following controls:

- "Forme du bouton" : A dropdown menu showing "Rectangle".
- "Sens bouton incurvé" : A dropdown menu showing "à gauche".
- "Rayon de courbure des coins du bouton rectangulaire" : A text input field containing the number "8", followed by the unit "Pixels".

At the bottom right of the dialog, there are three buttons: "Aide", "Annuler", and "OK".

Button Shape

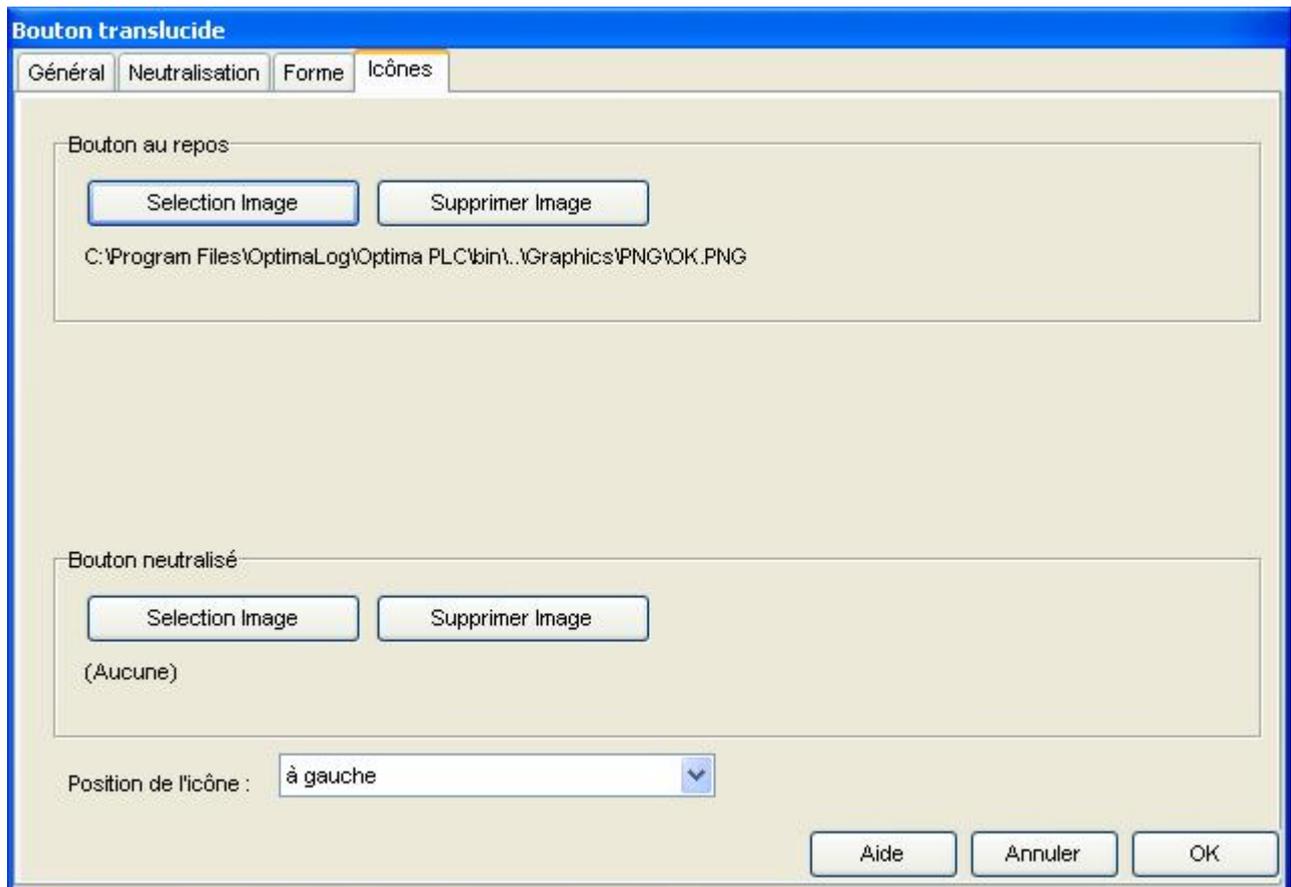
Select here the shape of the button : rectangle or curved

Curved Button direction

If the curved shape has been selected above, the user can specify the direction of the curve : left or right.

Corner's ray of curvature

This input box is enabled only if the shape of the button is rectangle. This parameter specifies the ray of curvature of the button rounded corners in pixels.

Dialog box "Translucent Function Button" - Icons :

Select here all the pictures to display on the function button. All formats are available, the PNG format is recommended. At the first picture selection, the open directory is by default the PNG Directory that contains a library of pictures in the PNG format.

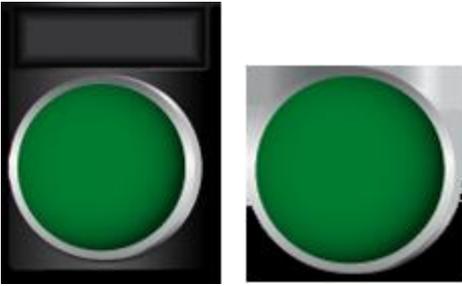
It's also possible to specify the picture position on the button in the dropped list "Icon Position"

3.2.6.2.7 Bouton poussoir industriel

Le bouton poussoir industriel représente un bouton type TELEMECANIQUE.

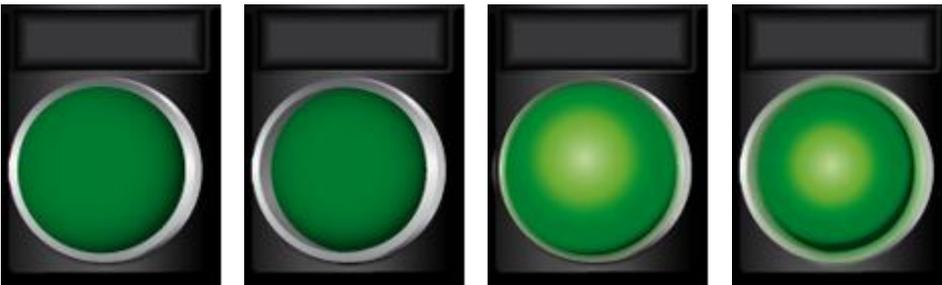
On peut définir une expression pour la valeur du bouton (enfoncé ou relâché) et également de manière facultative une expression pour la lampe associée au bouton. Dans ce cas, l'expression de lampe allumera le bouton quand sa valeur vaudra 1.

Le bouton industriel peut être avec ou sans étiquette (attention : il faut choisir l'objet correspondant dans la barre d'outils !!)



Le bouton peut donc avoir 4 aspects différents :

- relâché éteint
- relâché allumé
- enfoncé éteint
- enfoncé allumé



Boîte de dialogue de l'objet "Bouton Poussoir Industriel " - Onglet "Général" :

Bouton poussoir industriel

Général | Neutralisation

Expression bouton  

Expression lampe  

Couleur ▼

Aperçu



Type de bouton

Bistable

Monostable

Bouton "à accroche"

Etiquette

Texte Etiquette (Ligne N°1)

Afficher 2 lignes sur l'étiquette

Texte Etiquette (Ligne N°2)

Les champs de saisie de cette boîte sont les suivants :

Expression

Indiquer dans ce champ le nom de la variable à contrôler par le bouton. Cette variable doit être de type booléen

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Les champs de saisie de cette boîte sont les suivants :

Expression lampe

Indiquer dans ce champ le nom de la variable qui va définir l'état de la LED dans le bouton.

Couleur

5 couleurs sont disponibles : Vert, Blanc, Bleu, Jaune et Rouge.

Type de bouton

Cette option permet d'indiquer si le bouton est monostable ou bistable. S'il est monostable, il revient lorsque l'on cesse de le presser.

S'il est bistable, un clic positionnera le bouton enfoncé, il faut alors cliquer une seconde fois sur le bouton pour le relâcher.

Option "Bouton à accroche"

Si cette option est activée, appuyer sur le bouton activera la variable booléenne de l'expression associée. On ne pourra alors pas désactiver la variable en appuyant à nouveau sur le bouton.

Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

Boîte de dialogue de l'objet "Bouton Poussoir Industriel " - Onglet "Neutralisation" :

Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton poussoir sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

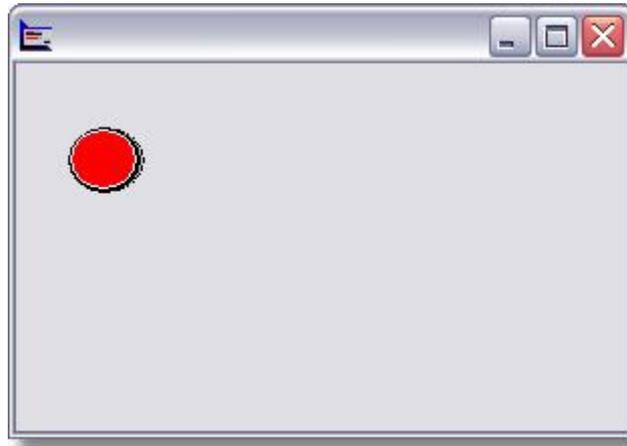
Option « Visible si neutralisé »

Si cette option est cochée, le bouton poussoir sera visible quand l'objet sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

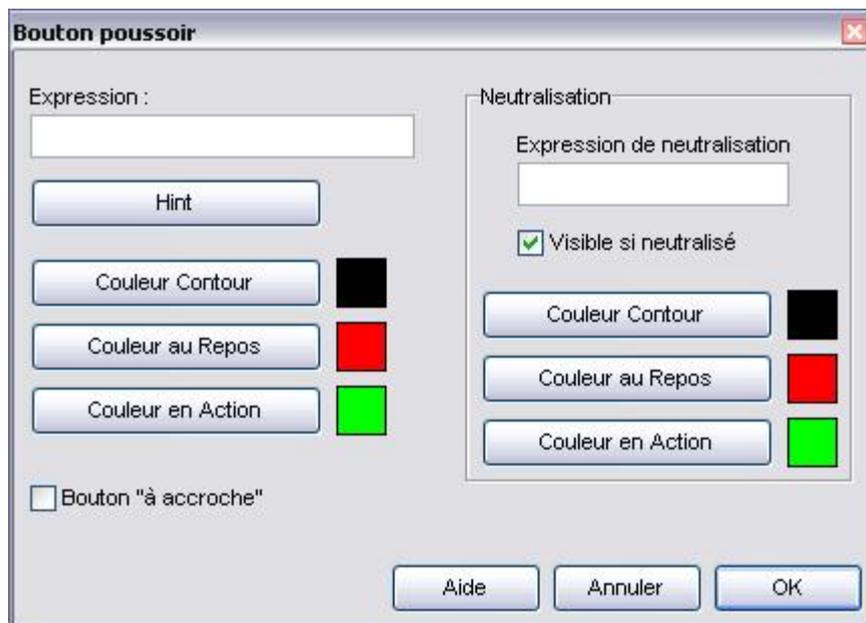
Si cette option n'est pas cochée, l'objet sera invisible quand il sera neutralisé.

3.2.6.2.8 Push button

This object represents a push button that permits to control a PLC variable or expression which type is Boolean.



Dialog box " Push Button " :



The parameters are :

Expression

Enter here the PLC variable or expression controlled by the push button (the variable type must be Boolean)

Exemple : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

"Button to hangs" Option

If this option is checked, with the first click on the button, the PLC variable will set to True. The user won't be able to reset the button by clicking on it. The button will only be relaxed when the PLC variable will be reset to FALSE in the PLC

Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The push button will be neutralized when the PLC variable will be set to True.

The user can set some colors to the push button when this one is neutralized (available only if the "Visible if neutralized Option" is checked

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the button will not be visible when neutralized.

If this option is checked, the object will be visible but the user won't be able to control it when neutralized.

3.2.6.3 INPUT BOX tab objects

3.2.6.3.1 Standard input box

This object is a standard input box which controls an alphanumeric PLC variable or expression. The user can input new values for the PLC variables with the keyboard.

Dialog box "Standard Input Box" - General :

The dialog box 'Objet de saisie' has three tabs: 'Général', 'Neutralisation', and 'Clavier virtuel'. The 'Général' tab is active. It features an 'Expression' text box containing 'IWR11' and an 'OPC' icon. Below it are buttons for 'Couleur fond' and 'Caractères'. There are two checked checkboxes: 'Taille verticale automatique' and 'Saisie autorisée'. An unchecked checkbox is 'Affichage format Time/Date Optima PLC'. Below it, 'Type Time Optima PLC' is set to 'time' and 'Format d'affichage' is set to 'dd/mm/yyyy" "hh:nn:ss'. A 'Hint' button is also present. On the right, the 'Valeurs numériques' section has input fields for 'Valeur min', 'Valeur max', 'Nb décimales' (value: 0), and 'Coefficient d'affichage'. A checked checkbox is 'Supprimer décimales superflues'. At the bottom right are 'Aide', 'Annuler', and 'OK' buttons.

The parameters are

Expression

Enter here the PLC variable or expression controlled by the input box. The PLC variable can be of any type : Integer, string,...

Example : R10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Min Value

If the type of the PLC variable defined in the expression field is numerical (Integer, real, ...), the min value specify an admissible lower limit when the user input a new value in the input box. If this limit is not respected , a message will advise the user that the entered value is not allowed. The min value can be a constant value or a PLC variable.

Example :

100
R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Max Value

If the type of the PLC variable defined in the expression field is numerical (Integer, real, ...), the max value specify an admissible upper limit when the user input a new value in the input box. If this limit is not respected , a

message will advise the user that the entered value is not allowed.
The max value can be a constant value or a PLC variable.

Example :

100
R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Decimal digits number

If a decimal numeric variable is displayed in input box, the user can specify the number of digits displayed after the decimal point.

"Suppress the superfluous decimals" Option

If a decimal numeric variable is displayed in the input box, this option when checked avoid to display the last digits after the decimal point if these digits are equal to 0.

Background color

Define here the background color of the input box.

Font

Select here the font of the text displayed in the input box. The "Font" button opens the font dialog box.

"Automatic vertical size" option

Automatically adapt the height of the input box according to the font (size) defined for the input box.

"Input allowed" option

If this option is not checked, the user will not be allowed to modify the value in the input box. The input box will only display the value of the PLC variable.

"OPTIMA PLC Time/Date display " option

Check this option only if the PLC variable is valued on OPTIMA PLC and the PLC variable type is time, date, date_and_time or time_of_day.

In this case, the variable will be displayed in a date/time format into the input box.

Example :

12/02/2008 22:10:45

OPTIMA PLC Time type

Specify here the time type of the OPTIMA PLC variable to display. The following choices are available :

Time, Date, Date_and_time, Time_of day

Display format

Select here the format used to display Dates / times in the input box.

It is possible to select a preselected format in the dropped list of the field or to enter a special format.

Examples of format :

dd / mm / yyyy" --- "hh:nn:ss displays 02/03/2002---18:23:15

AM / PM hh:nn displays 6:23PM

"d" dddd" "mm" "yyyy displays Saturday March 2002 2

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

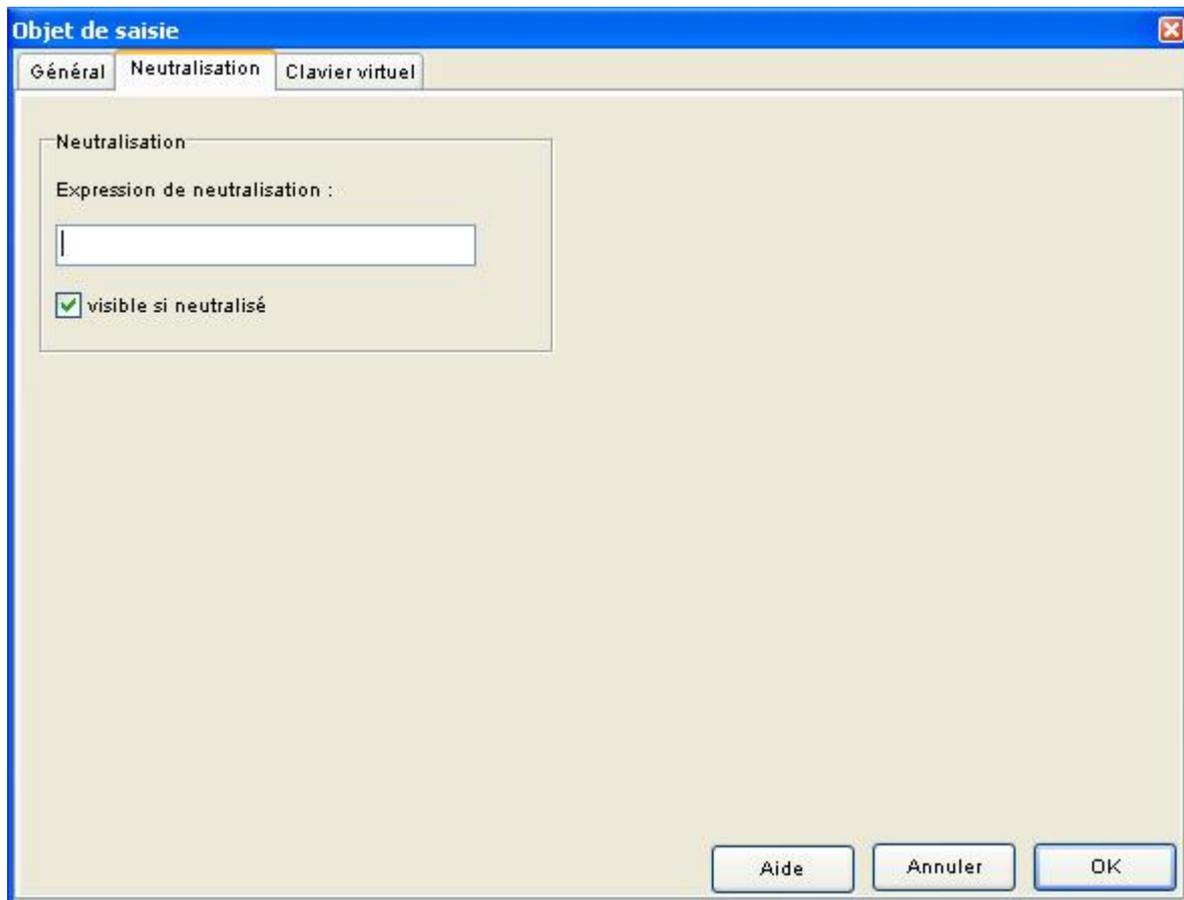
« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Dialog box "Input Box" - Neutralization



Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The input box will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

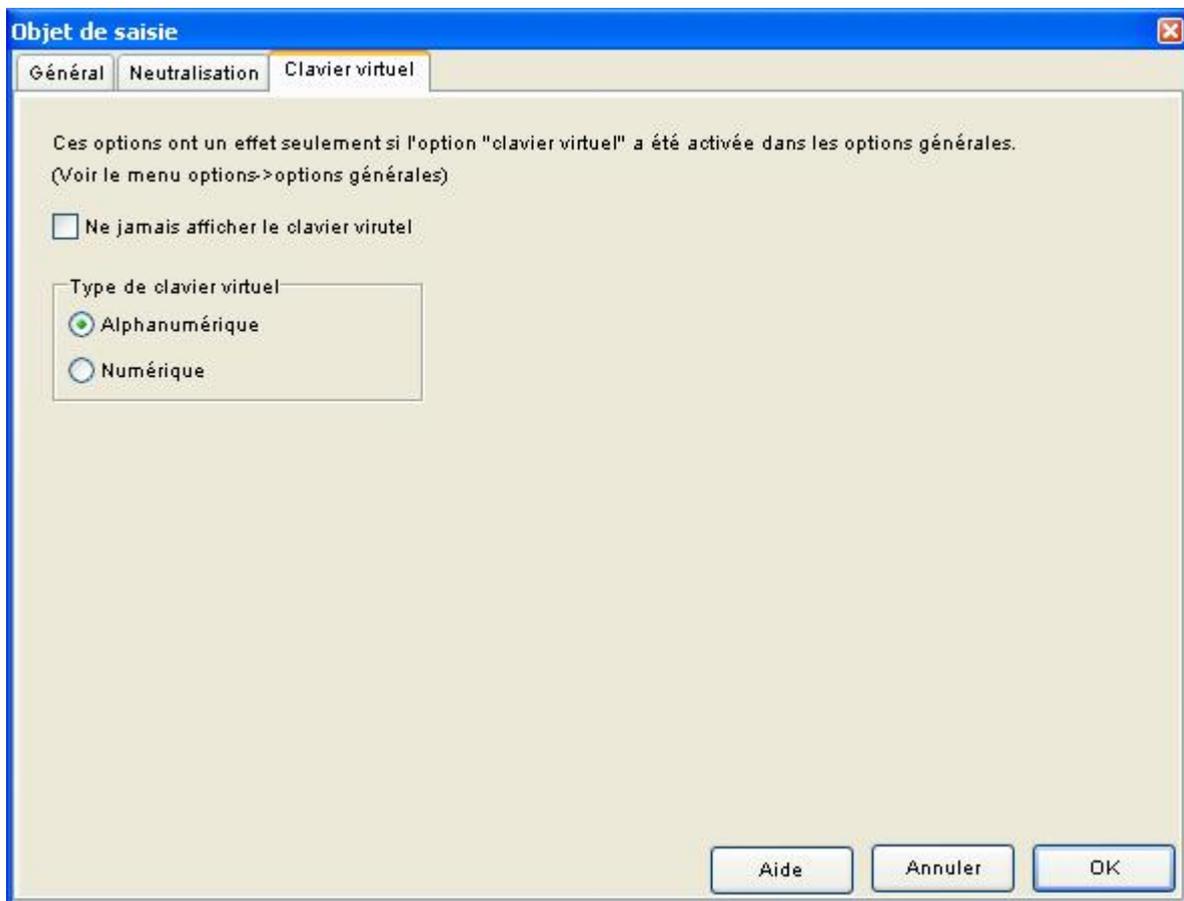
It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the input box will not be visible when neutralized.

If this option is checked, the input box will be visible but the user won't be able to input some values.

Dialog box "Input box" - Virtual Keyboard



The virtual keyboard is used on tactile screens that are not fitted with a keyboard. It is a virtual keyboard that opens up to the screen when the user click on the input box. This virtual keyboard permits to the user to enter alphanumeric characters by clicking with the mouse on the buttons of this keyboard.

To activate the automatic virtual keyboard in your project, use the menu "Option" then "general Options" and check the "automatic virtual keyboard" option.

The virtual keyboard parameters will only have an effect if the option "Virtual Keyboard" option is checked.

"No virtual keyboard display" option

If this option is checked, the virtual keyboard will not be displayed for this input box even if the "Virtual Keyboard" option is checked in the project.

" virtual keyboard" type

The user can choose between two types of virtual keyboard : Alphanumerical or numerical. The default virtual keyboard type is alphanumerical

Example : Alphanumerical virtual keyboard

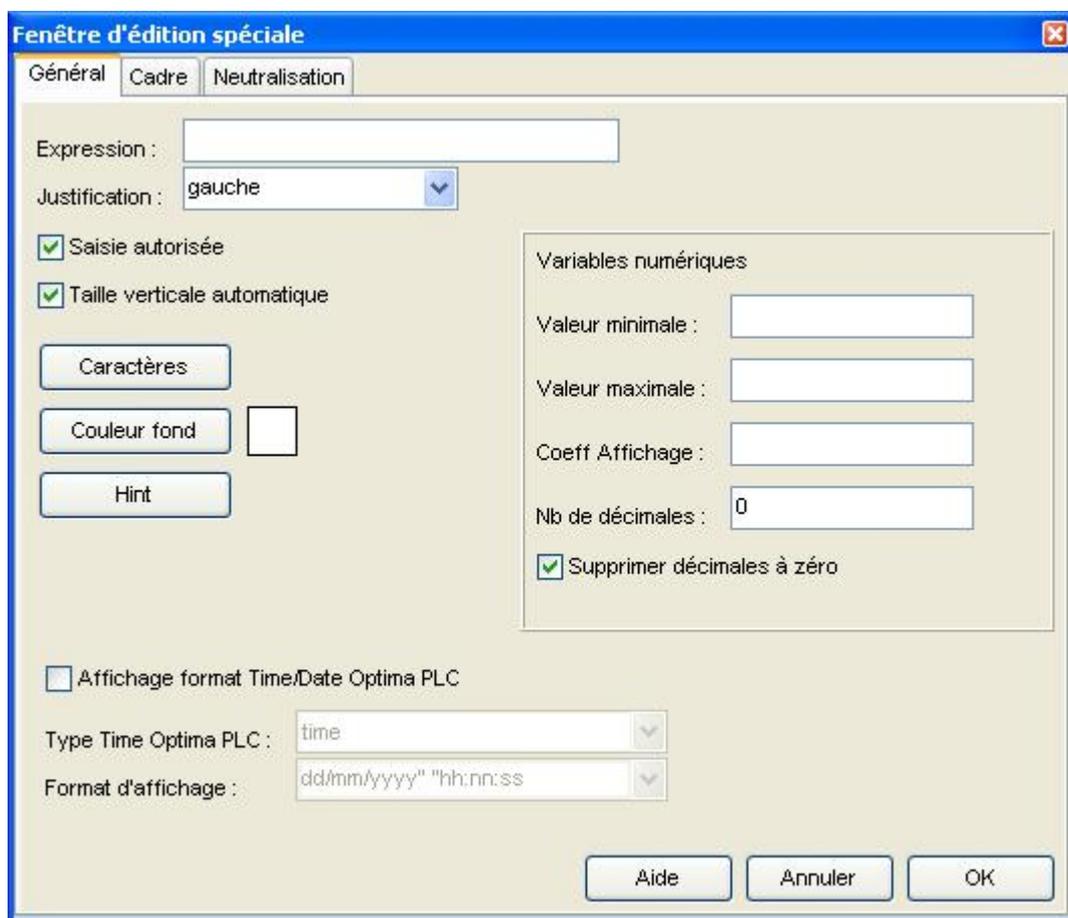


3.2.6.3.2 Custom input box

This object is a standard input box which controls an alphanumeric PLC variable or expression. The user can input new values for the PLC variables with the keyboard.

The custom input box has more functionalities than the standard input box like the alignment option of the text: Left, centered or on the right.

Dialog box " Custom input box " : General



The parameters are :

Expression

Enter here the PLC variable or expression controlled by the input box. The PLC variable can be of any type : Integer, string,...

Example : R10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Justification

Select here the alignment of the text inside the input box, the available choices are :
On the left, centered, on the right.

"Input allowed" option

If this option is not checked, the user will not be allowed to modify the value in the input box. The input box will only display the value of the PLC variable.

"Automatic vertical size" option

Automatically adapt the height of the input box according to the font (size) defined for the input box.

Background color

Define here the background color of the input box.

Font

Select here the font of the text displayed in the input box. The "Font" button opens the font dialog box.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

"OPTIMA PLC Time/Date display " option

Check this option only if the PLC variable is valued on OPTIMA PLC and the PLC variable type is time, date, date_and_time or time_of_day.

In this case, the variable will be displayed in a date/time format into the input box.

Example :

12/02/2008 22:10:45

OPTIMA PLC Time type

Specify here the time type of the OPTIMA PLC variable to display. The following choices are available :
Time, Date, Date_and_time, Time_of day

Display format

Select here the format used to display Dates / times in the input box.

It is possible to select a preselected format in the dropped list of the field or to enter a special format.

Examples of format :

dd / mm / yyyy" --- "hh:nn:ss displays 02/03/2002---18:23:15

AM / PM hh:nn displays 6:23PM

"d" dddd" "mm" "yyyy displays Saturday March 2002 2

Min Value

If the type of the PLC variable defined in the expression field is numerical (Integer, real, ...), the min value specify an admissible lower limit when the user input a new value in the input box. If this limit is not respected, a message will advise the user that the entered value is not allowed.

The min value can be a constant value or a PLC variable.

Example :

100

R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Max Value

If the type of the PLC variable defined in the expression field is numerical (Integer, real, ...), the max value specify an admissible upper limit when the user input a new value in the input box. If this limit is not respected, a message will advise the user that the entered value is not allowed.

The max value can be a constant value or a PLC variable.

Example :

100

R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Decimal digits number

If a decimal numeric variable is displayed in input box, the user can specify the number of digits displayed after the decimal point.

"Suppress the superfluous decimals" Option

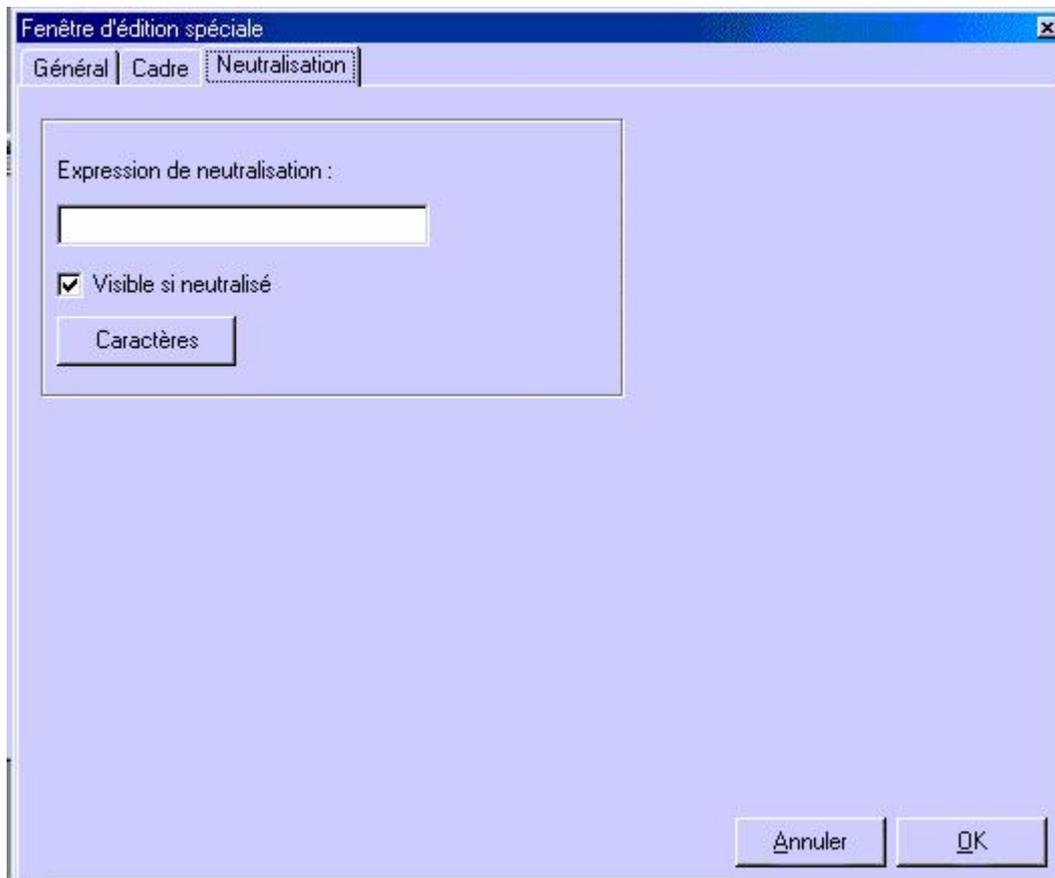
If a decimal numeric variable is displayed in the input box, this option when checked avoid to display the last digits after the decimal point if these digits are equal to 0.

Dialog Box "Custom input box" : Frame

"Display bevel" option

If this option is not checked, the window will be displayed with no frame.

Dialog box " Custom input box " : Neutralization



Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The input box will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

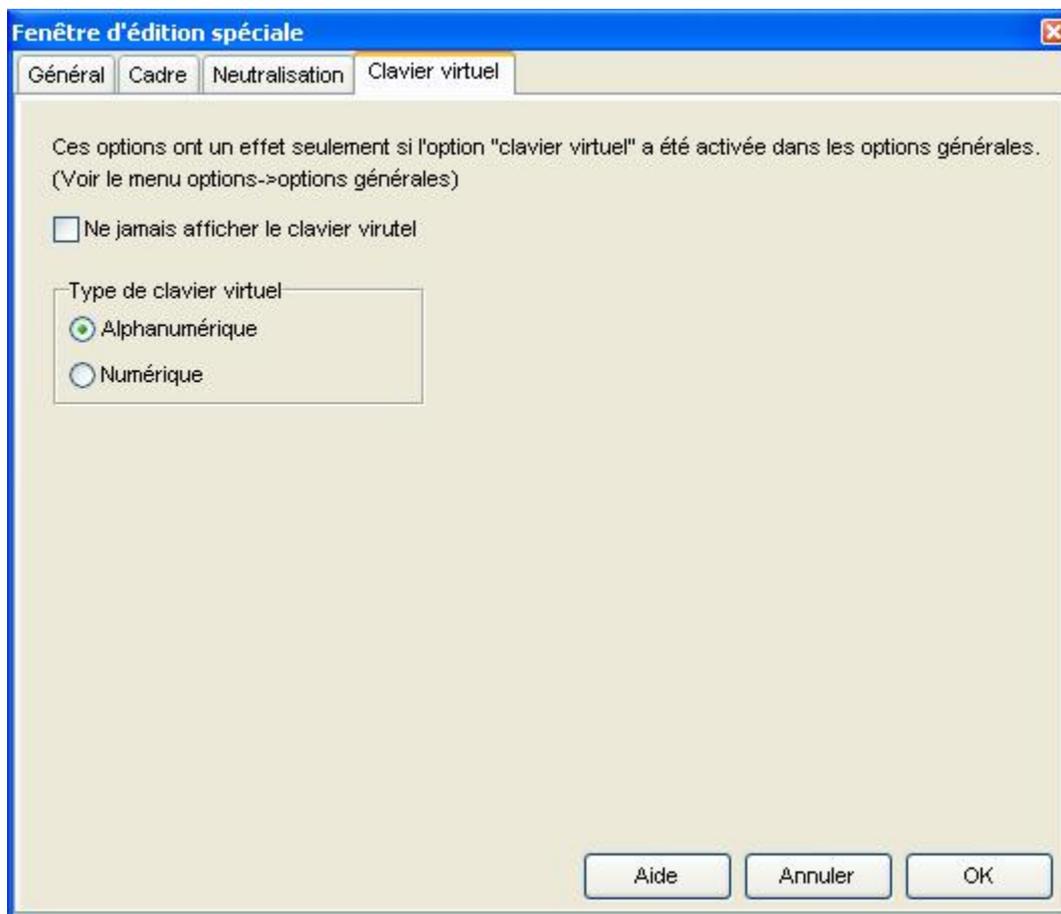
If this option is not checked, the input box will not be visible when neutralized.

If this option is checked, the input box will be visible but the user won't be able to input some values.

" Font " Button

The user can specify a font to be assigned to the input box text when neutralized.

Dialog Box "Custom Input Box" - Virtual Keyboard



The virtual keyboard is used on tactile screens that are not fitted with a keyboard. It is a virtual keyboard that opens up to the screen when the user click on the input box. This virtual keyboard permits to the user to enter alphanumeric characters by clicking with the mouse on the buttons of this keyboard.

To activate the automatic virtual keyboard in your project, use the menu "Option" then "general Options" and check the "automatic virtual keyboard" option.

The virtual keyboard parameters will only have an effect if the option "Virtual Keyboard" option is checked.

"No virtual keyboard display" option

If this option is checked, the virtual keyboard will not be displayed for this input box even if the "Virtual Keyboard" option is checked in the project.

" virtual keyboard" type

The user can choose between two types of virtual keyboard : Alphanumerical or numerical. The default virtual keyboard type is alphanumerical

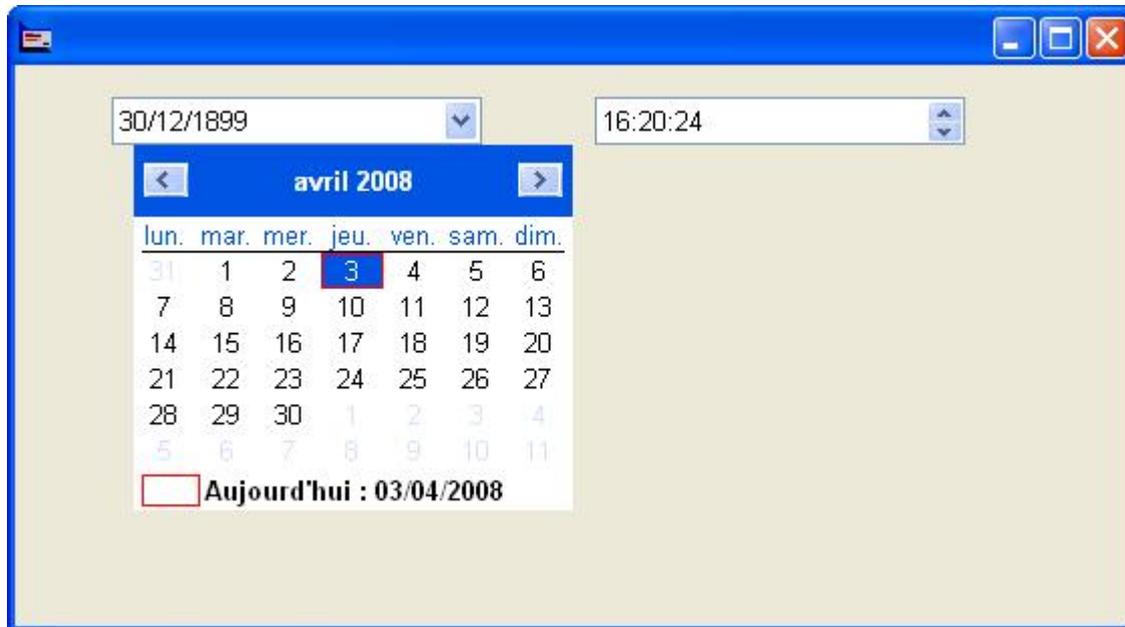
Example : Alphanumerical virtual keyboard



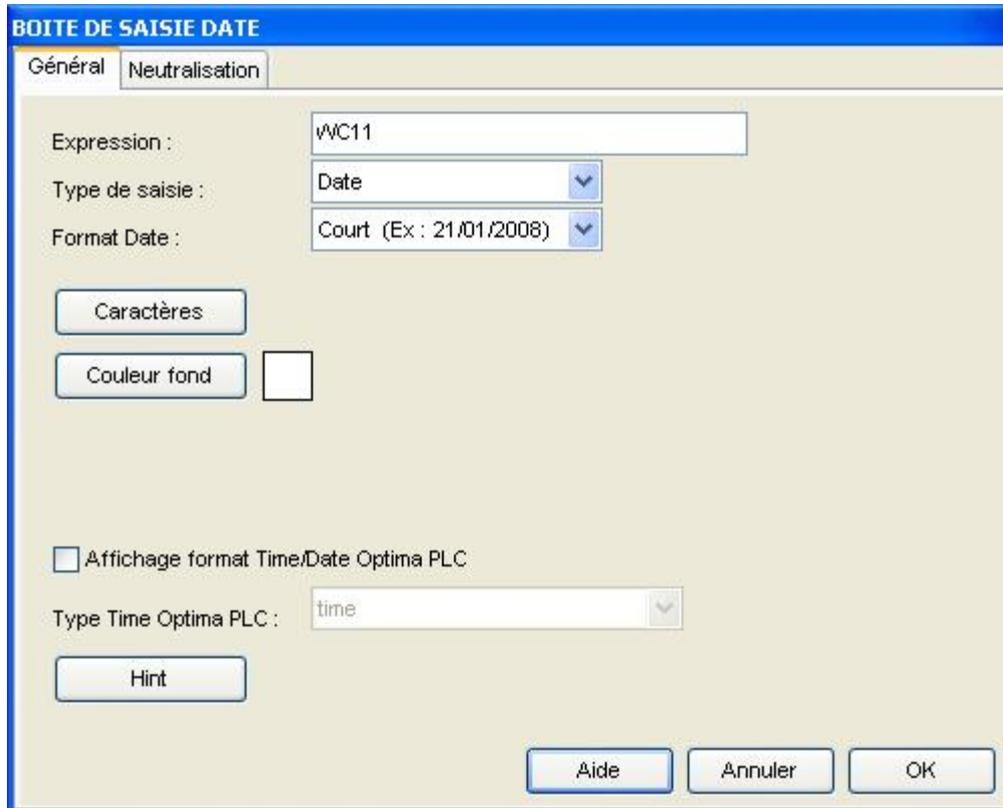
3.2.6.3.3 Date / Time input box

This object is a special input box used to display and input date/time values. It can control some PLC variables or expression which type is string (C12 for example) coming from APIGRAF/OPTIMA PLC or time, date, date_and_time, time_of_day coming only from optima PLC.

When the type of input selected is date, a calendar opens up automatically to allow the user to choose a new date to affect to the PLC variable.



Dialog Box "Date / Time Input Box" - General



The parameters are :

- **Expression**

Enter here a PLC variable or expression of **type string coming from APIGRAF/OPTIMA PLC or of type time,**

date, date_and_time, time_of_day coming from optima PLC.

This variable will be affected by the date / hour selected in the month calendar. If the value of the variable is updated in the PLC, the value displayed in the input box with the date/time format will also be updated

Example : C10
WC10
date1

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Input box type

Two types are available : Date or hour.

With the date type, a month calendar opens up automatically each time the user wants to modify the value.

Dates Format

This field is only available when the type date is selected. The user can specify the date format :

Short format: 21/02/2008

Long format: Saturday February 2, 2008

Font

Select here the font of the text displayed in the input box. The "Font" button opens the font dialog box.

Background color

Define here the background color of the input box.

"OPTIMA PLC Time/Date display " option

Check this option only if the PLC variable is valued on OPTIMA PLC and the PLC variable type is time, date, date_and_time or time_of_day.

In this case, the variable will be displayed in a date/time format into the input box.

OPTIMA PLC Time type

Specify here the time type of the OPTIMA PLC variable to display. The following choices are available : Time, Date, Date_and_time, Time_of day

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Dialog Box " Date/Time input Box" - Neutralization

Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The input box will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the input box will not be visible when neutralized.

If this option is checked, the input box will be visible but the user won't be able to input some values.

3.2.6.3.4 Combo box

This object is a input box with a dropped list with user-defined items.

The user can input a value with the keyboard or select an existing valuer in the dropped list associate to the input box.

Dialog Box "Combo box" - General

The screenshot shows a dialog box titled "Edition avec liste déroulante" with a blue title bar and a close button. It has three tabs: "Général", "Neutralisation", and "Clavier virtuel". The "Général" tab is selected. The dialog contains the following elements:

- An "Expression :" label followed by a text input field.
- A "Caractères" button below the expression field.
- A "Liste du menu déroulant :" label followed by a large empty list box.
- Two checkboxes:
 - Affectation de l'index de l'élément à l'expression
 - Saisie autorisée
- A "Hint" text input field.
- Three buttons at the bottom right: "Aide", "Annuler", and "OK".

The parameters are:

" Expression "

To indicate in this field the name of the variable to control. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Combo item listing"

To specify here the list of the accessible possible choices in the list associated to the window of edition. To go to the line to separate every element of the list.

Example :

Choix1
Choix2
Choix3

It is possible to bring some references to parameters in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

" Assign the selected item index to the expression " Option

This option is not checked by default. If this option is not checked, the value of the element of the list that is affected to the expression.

If this option is checked, the item value selected will be affected to the expression.

Example :

Expression : WR100

List :

100
200
500
1000

If the user chooses the element 200 of the list then the value 200 is affected to the WR100 register

If the option "**Assign the selected item index to the expression**" is checked then if the user chooses 200 in the list, the value 1 will be affected to the WR100 register.

(100 are the element of index 0, 200 the element of index 1, 500 the element of index 2,). The value 1 corresponds to the selected item index in the dropped list.

"Input allowed" option

If this option is checked, the operator will be able to input a new value in the input box or to select a value in the dropped list.

If this option is not checked, the operator will be able to select a value in the dropped list.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

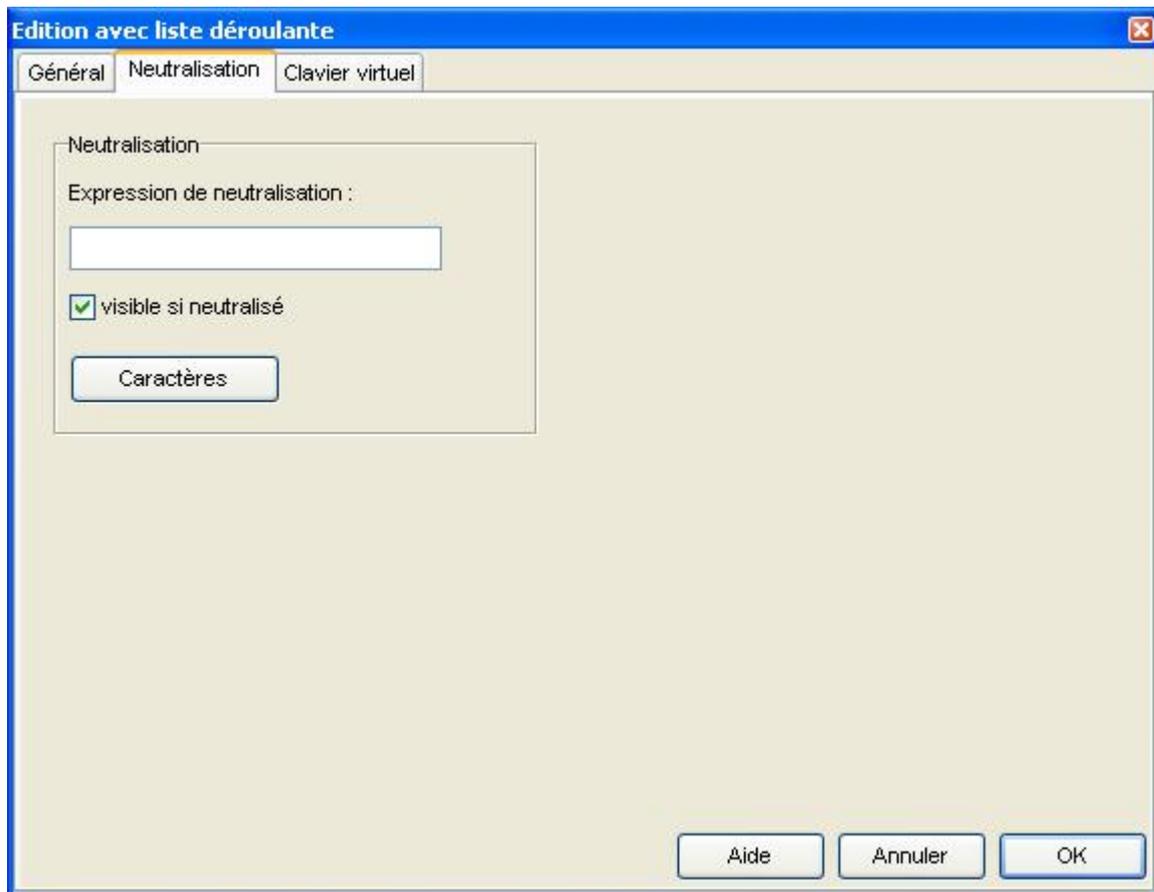
« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Dialog Box "Combo box" - Neutralization



Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The input box will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the input box will not be visible when neutralized.

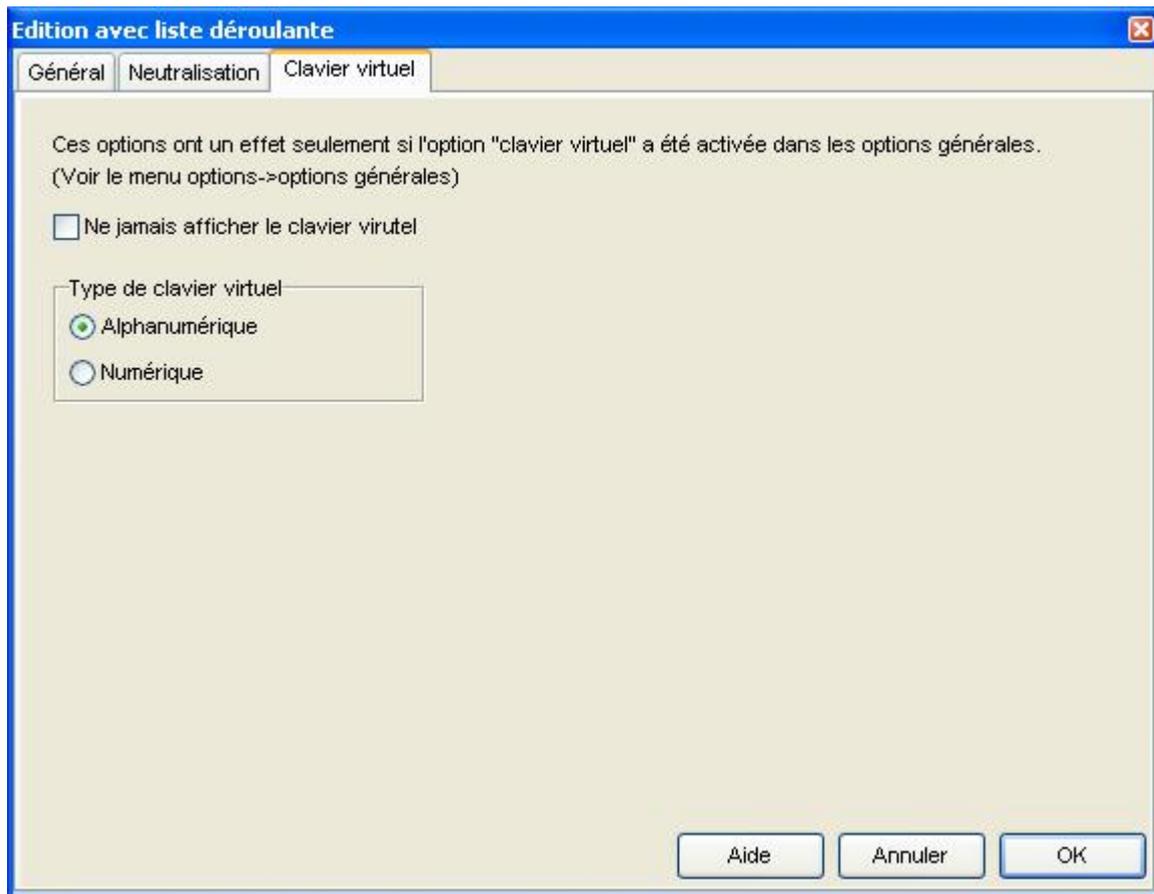
If this option is checked, the input box will be visible but the user will not be allowed to input a new value.

"Font" for " Neutralization "

A font can be defined for the neutralization. When neutralized, this font will be assigned to the input box text when

neutralized.

Dialog Box "Combo box" - virtual Keyboard



The virtual keyboard is used on tactile screens that are not fitted with a keyboard. It is a virtual keyboard that opens up to the screen when the user click on the input box. This virtual keyboard permits to the user to enter alphanumeric characters by clicking with the mouse on the buttons of this keyboard.

To activate the automatic virtual keyboard in your project, use the menu "Option" then "general Options" and check the "automatic virtual keyboard" option.

The virtual keyboard parameters will only have an effect if the option "Virtual Keyboard" option is checked.

"No virtual keyboard display" option

If this option is checked, the virtual keyboard will not be displayed for this input box even if the "Virtual Keyboard" option is checked in the project.

" virtual keyboard" type

The user can choose between two types of virtual keyboard : Alphanumerical or numerical. The default virtual keyboard type is alphanumerical

Example : Alphanumerical virtual keyboard

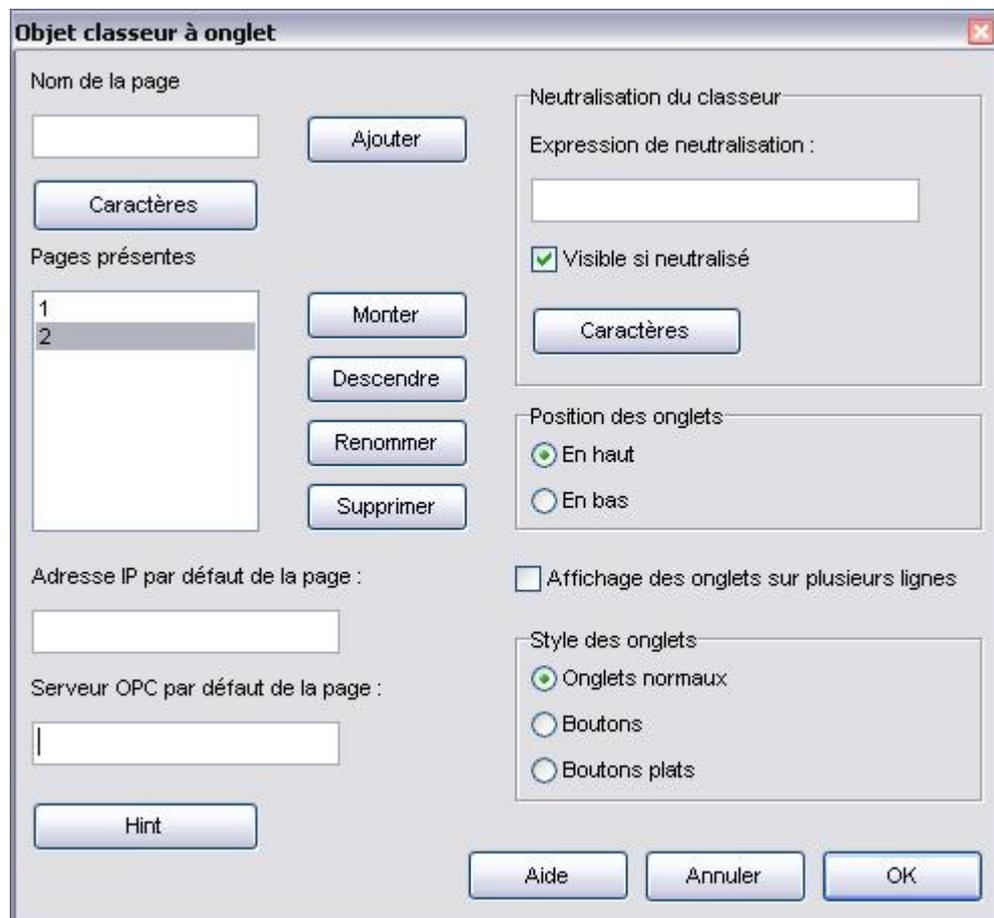


3.2.6.4 CONTROLS tab objects

3.2.6.4.1 Page control

This object is a page control with tabsheets.

Dialog box parameter "Page Control"



The parameters are :

Page Name

Enter here the name of the new tab sheet to add to the page control.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Add Button

The "add" button adds a new tab sheet with the page name if the page name field is not empty.

Pages list

This list shows all tab sheets included in the page control. The user can select a tab sheet in the list. If a tab sheet is selected, it will be displayed in designing mode.

"Up" button

Moves up the selected tab sheet in the tab sheet list.

"Down" button

Moves up the selected tab sheet in the tab sheet list.

"Rename" button

If one tab sheet is selected in the list, use the "rename" button to modify the page name of an existing tab sheet.

"Suppress" button

Suppress the selected tab sheet in the list.

Font

Select here the font used to display the tab sheet titles. Opens the font dialog box.

Default IP address :

The user can define a default IP address of the page control. If a default address has been defined, all objects belonging to the page control will use this IP address (unless an IP address is defined to the lower level) for their update.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Default OPC server name

The user can define the default OPC Server name. If a default OPC Server name has been defined, all objects belonging to the page control will use this OPC Server for their update.

Caution : if an IP address is defined, the default OPC server will not be taken in account.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The page control will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the page control will not be visible when neutralized.

If this option is checked, the page control will be visible but the user will not be allowed to select a tab sheet in the page control.

More over, the objects contained in the tab sheets will also be disabled.

" Font " Button for " Neutralization "

The user can select a special font assigned to the tab sheet titles when the page control is neutralized. This font is only available is the "Visible if neutralized" option is checked.

Tab controls position

Select here if the tabs must be at top or at the bottom of the page control.

" Tab Control multi line display" Option

When too much tab sheets are defined, they are not able to all to be displayed. If this option is checked, all tabs will be displayed on several lines. (as any as required)

Tab Control style

Select here the style of the tabs :

Normal tabs, buttons or flat buttons.

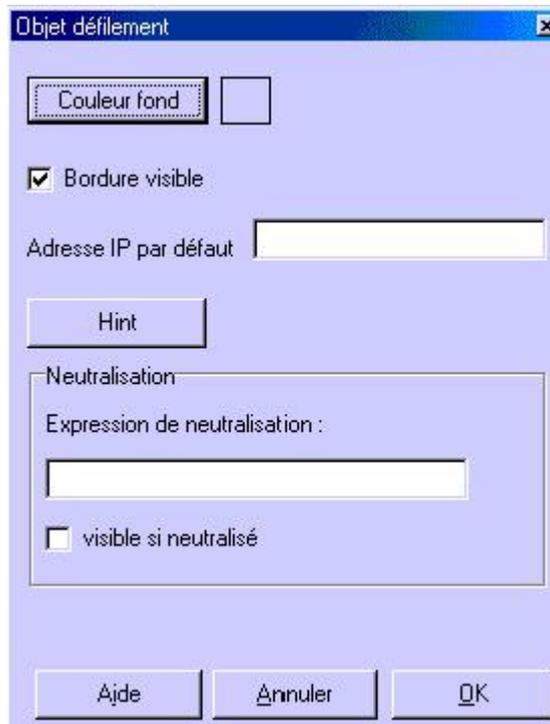
3.2.6.4.2 Scroll box

The object "Scroll box" is like a Panel that can be scrolled using the scrolling bars with the mouse. The object is able to contain other object and the position of the objects into the scroll box can be out of the scroll box size limits.

If the size of the objects contained in the Scroll box object is more important than the object scrolling itself, scroll bars are created automatically on the bottom and on the right of the scroll box.

If one wants to include some new objects in a scroll box, it is necessary to create these objects inside the scroll box area.

Dialog box "Scroll Box"



The options are the following :

Background color

Select here the background color of the scroll box.

"Display frame" option

If this option is not checked, the border of the scroll box object will not be visible.

Default IP address :

The user can define a default IP address of the scroll box. If a default address has been defined, all objects belonging to the scroll box will use this IP address (unless an IP address is defined to the lower level) for their update.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The scroll box will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the scroll box will not be visible when neutralized.

If this option is checked, the scroll box will be visible but the user will not be allowed to scroll into the scroll box. More over, the objects contained in scroll box will also be disabled.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

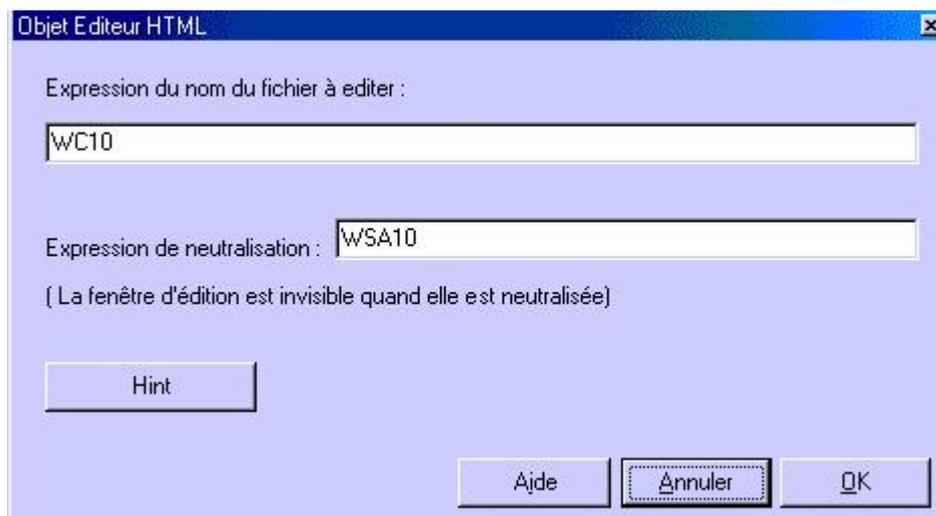
Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.4.3 HTML Viewer

This object is used to edit a simple HTML file.

The name of the HTML file can be a constant string or a PLC variable / expression of string type.

Dialog Box " HTML Viewer"



The parameters are :

"File name expression" to edit :

This parameter gives the name of the file HTML to publish. It can be a constant string :
'C:\README.HTML'

In this case, the file name must be entered between single quotes.

It also be a PLC Variable of string type :

C30

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !C30

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

In this case, to every change of the value of C30, the file whose name is the value of C30 will be displayed.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.4.4 Panel

The object is a panel which can contain other objects. The user can also display a title on the panel.

Dialog Box "Panel"

The parameters are :

Text

Enter here a text value corresponding to the title displayed on the panel. This field can also be empty if the panel is used to contain other objects.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Text Position

This option permits to choose the justification of the text in the object, the possible choices are :

- Left Justified : the text will be aligned on the left in the panel
- Centered: the text will be centered in the panel.
- Right Justified : the text will be aligned on the right in the panel.

Font

Select here the font to use for the title displayed " Panel ". Opens the font dialog box. The font selected will also be the default font of the panel. Each object inserted in the panel will be created with this default font. The user can then modify the object's font.

Background color

Select here the background color of the Panel.

"Display frame" option

If this option is not checked, the border of the panel will not be visible.

In Frame type

The user can choose an internal frame, the different styles available are :

- None

- In hollow
- In relief

Out Frame type

The user can choose an external frame, the different styles available are :

- None
- In hollow
- In relief

Frame thickness (in pixel)

Defines the panel border thickness. The default value is 1 pixel.

Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The panel will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the panel and all objects in the panel will not be visible when neutralized. If this option is checked, the panel will be visible but objects contained in the panel will also be disabled.

Default IP address :

The user can define a default IP address of the panel. If a default address has been defined, all objects belonging to the panel will use this IP address (unless an IP address is defined to the lower level) for their update.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Default OPC server name

The user can define the default OPC Server name for the panel. If a default OPC Server name has been defined, all objects belonging to the panel will use this OPC Server for their update.

Caution : if an IP address is defined, the default OPC server will not be taken in account.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« **Hint** »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.4.5 List Box

The list box displays a list of items. The user can select an item in the list.
Example of listbox object :



A PLC Variable or expression can be controlled by the list box object.
There different ways to use the list box object

- Simple texts list : a list of texts is defined in the object. The user can select an item (text) in the list. The index of the selected item is affected to the PLC variable value. (The index available values are 0,...ListCount -1)
- Expressions list : a list of PLC variables/expression is defined in the object. The PLC variable values are added to the list. The index of the selected item is affected to the PLC variable value. (The index available values are 0,...ListCount -1)

Example : Expressions list : R10,R12,R14 avec (R10=1,R12=3 et R14=15) ,the list will display 3 items : 1,3,15.

If the main PLC variable of the list box is R50 then :

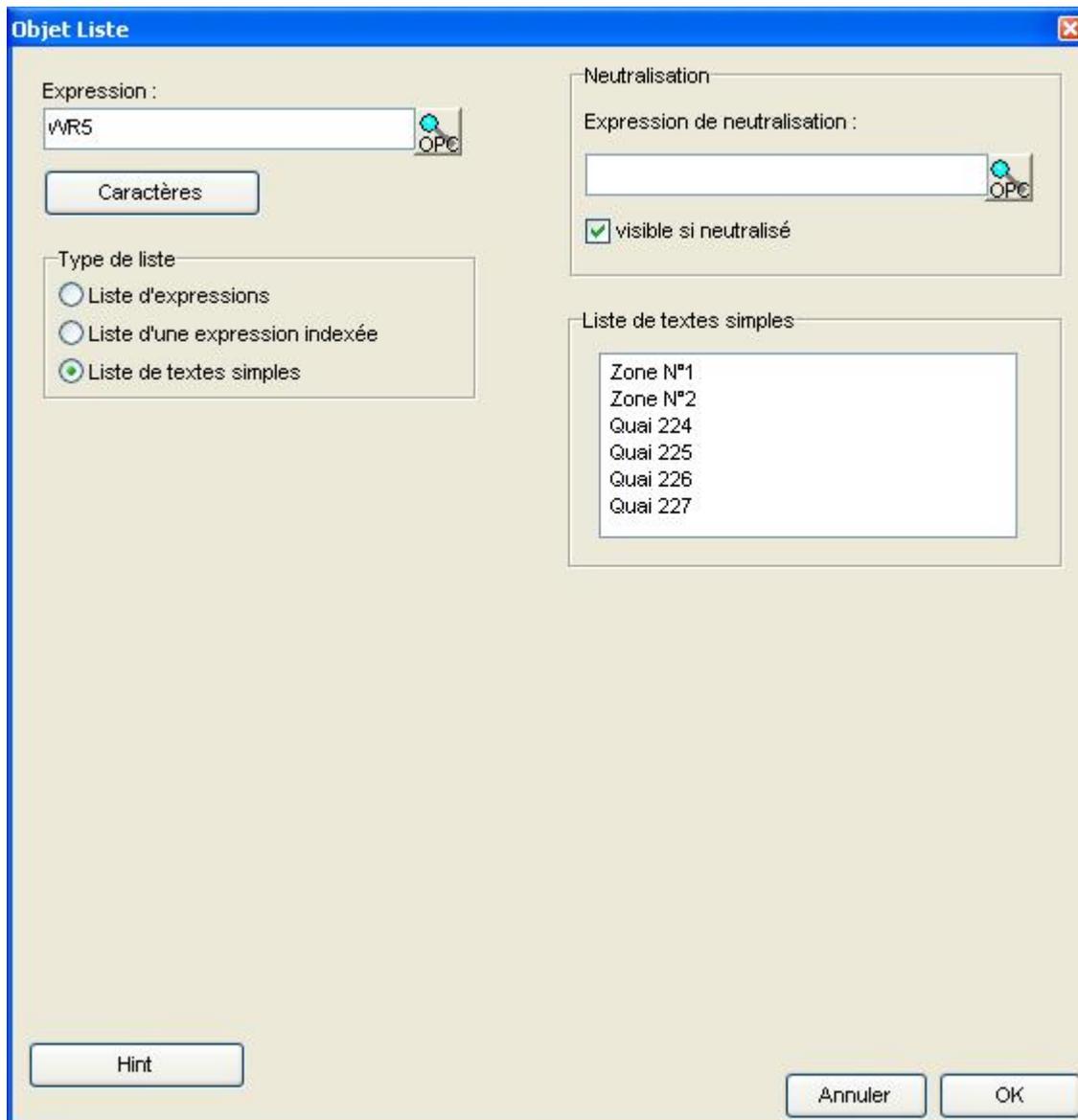
When the user click on the value 1, 0 is affected to R50

When the user click on the value 3, 1 is affected to R50.

When the user click on the value 15, 2 is affected to R50

- Indexed expressions list : This control mode is the near than the "Expressions list" mode. The difference is that an indexed PLC variable is defined in the list box instead of an expressions list. The indexed PLC variable is for example : C[*] , The symbol "*" with the brackets ([*]) is put at the index position . The Start and the End index are also defined. Example (StartIndex : 0, EndIndex : 19). In this example the values of C0,C1,C2,..C19 are listed in the project. As the mode above, the index of the selected item is affected to the PLC variable value. (The index available values are 0,...ListCount -1)

Dialog box "List Box" - Simple texts list



The parameters are :

Expression

Enter here the PLC variable or expression controlled by the list box. The PLC variable can be of type : Numerical or string.

Example : R10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Font

Select here the font used to display the texts of the radio group radio

List Type

Select here the list mode type (See above about the different list modes). Depending on the list type selected, a memo is displayed to enter the list items (Enter one item by line)

With the list type "Indexed expressions list", 3 input boxes are displaying to input the indexed expression (Example : C[*]), the start index (Example : 0 or 1) and the end index (Example : 20).

Neutralization Expression

Enter here the PLC variable or expression of neutralization (The variable type must be Boolean). The list box will be neutralized when the PLC variable will be set to True.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Visible if neutralized " Option

If this option is not checked, the list box will not be visible when neutralized.

If this option is checked, the list box will be visible but the user will not be allowed to select an item in the list.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.4.6 Click zone

The click zone is an area which is delimited in designing mode on a supervisory form. This area is not visible in execution mode.

The click zone can be used in several ways :

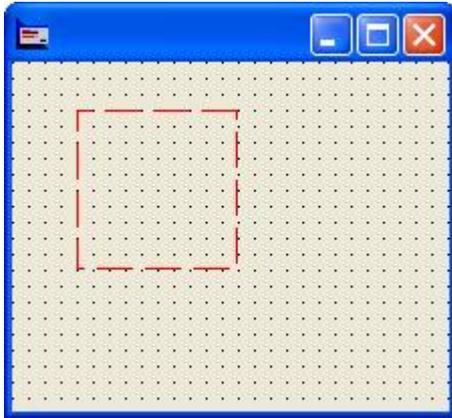
Execute a macro when the user make a left click down with the mouse in the click zone

Execute a macro when the user release the mouse button.

Control a PLC variable (Boolean) : In monostable mode, the variable is set to 1 when the mouse left button is down in the click zone and is reset when the mouse left button is released.

In bistable mode, the variable is set to 1 with the first mouse left click and reset with the next mouse left click.

In designing mode, the "Click zone" object looks like :



Dialog box "Click zone"

 A screenshot of a dialog box titled "ZONE DE CLICK". It has a blue title bar and a light beige background. The dialog contains the following elements:

- Expression:** A text input field with an "OPC" button to its right.
- Type d'activation:** A group box containing two radio buttons: "Bistable" (unselected) and "Monostable" (selected).
- Macro enfacement:** A large empty text area.
- Macro relâchement:** A large empty text area.
- Buttons:** A "Hint" button on the left, and "Annuler" and "OK" buttons at the bottom right.

The parameters are :

Expression (optionnal)

Enter here the PLC variable (boolean) to be controled by the click zone.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If the variable is valued on an OPC server, the "OPC" button opens the OPC server explorer which helps the user to find a valid tag on an OPC server installed on the PC.

The user can set the field "Expression" empty if he only wants to use macros with the click zone.

Example :The user wants to open the supervisory form "TEST" by clicking on the click zone. He only has to enter the following line in the field "release macros" : SUPERG(TEST)

Bistable or monostable

When an expression is defined, the user can choose between two activation mode for the boolean variable.

In monostable mode, the variable is set to 1 when the mouse left button is down in the click zone and is reset when the mouse left button is released.

In bistable mode, the variable is set to 1 with the first mouse left click and reset with the next mouse left click.

On Press Macro

Entrer la ou les macros à exécuter lors de l'enfoncement du bouton droit de la souris sur la zone de clic. Si il y a plusieurs macros, entrer une seule macro par ligne.

On Relaxed Macro

Entrer la ou les macros à exécuter lors du relachement du bouton droit de la souris.

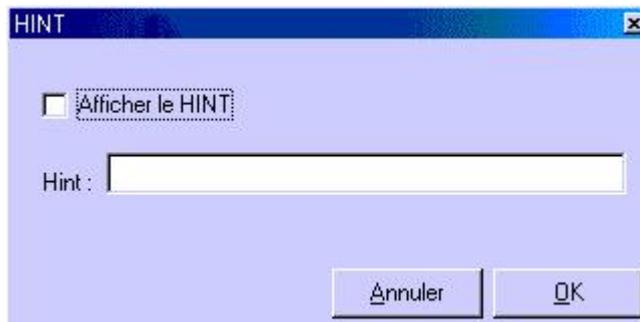
Si il y a plusieurs macros, entrer une seule macro par ligne.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

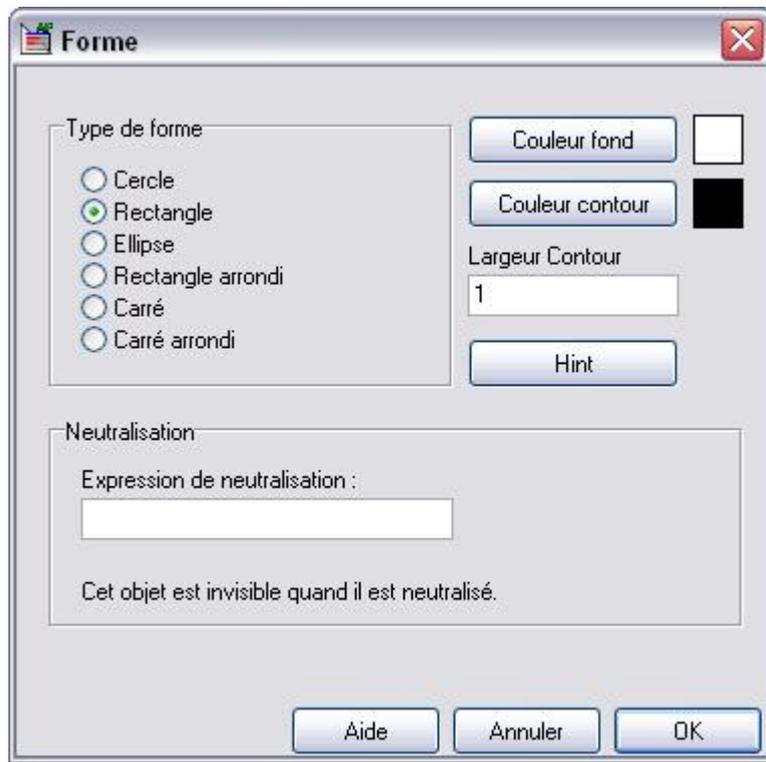
Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.5 DRAW tab objects

3.2.6.5.1 Shape

This object displays a graphic shape.

Dialog box Shape



The parameters are the following :

Shape type

Select here the type of shape, the different choices are :

- Circle
- Ellipsis
- Rectangle
- Rounded rectangle
- Square
- Square rounded

Background color

This option permits to adjust the background of the shape.

Frame color

This option permits to adjust the frame color of the shape.

Frame width (in pixels)

Specify the frame width of the shape, the default value is 1 pixel.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text.

Neutralization expression

Define a variable or expression for neutralization (The variable type must be Boolean). The shape will be invisible every time that the neutralization variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

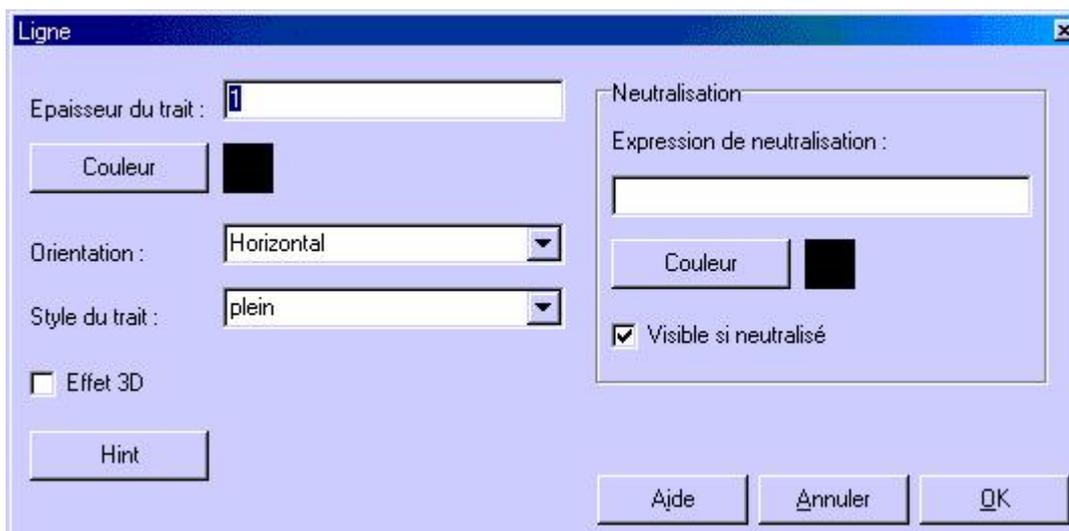
3.2.6.5.2 Line

This graphic object display a graphic line on a supervisory form.

The lines of thickness equal to 1 can be displayed with dotted lines, dashes,...

The lines of thickness superior to 1 can be displayed with a 3D effect, that gives a "tube like" display.

Dialog Box " Line "



The parameters are :

Line Thickness

Specifies the line thickness.

Caution : Only lines of thickness equal to 1 can be displayed in dotted lines or in dashes. The lines of thickness superior to 1 can be displayed with a 3D effect.

" Color " Button

Choose the line color. This button opens the color dialog box.

" Orientation "

Specify the line orientation. The orientation can be Horizontal, vertical, or oblique line.

" Line Style "

The lines of thickness equal to 1 can be displayed in dotted lines or in dashes.

" 3D Effect " option

If this option is checked and if the thickness of the line is superior to 1, then the line will appear with an effect 3D

that gives it a " Tube like " displaying.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Neutralization Expression

The user can specify the variable or the expression of neutralization (This one must be a Boolean one). The line will not be visible when the variable or the expression value will be set to TRUE if the option "Visible if neutralized" is checked.

If the option "Visible if neutralized" is not checked, the line will not be visible when neutralized.

Example : SA10

It's possible to specify the distant IP address of the APIGRAF/OPTIMA PLC workstation in the expression. In this case, this address will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Neutralization Color

The user can define the line color when this one is neutralized.

Caution : if the option "Visible if neutralized" is not checked, the line will not be visible when neutralized.

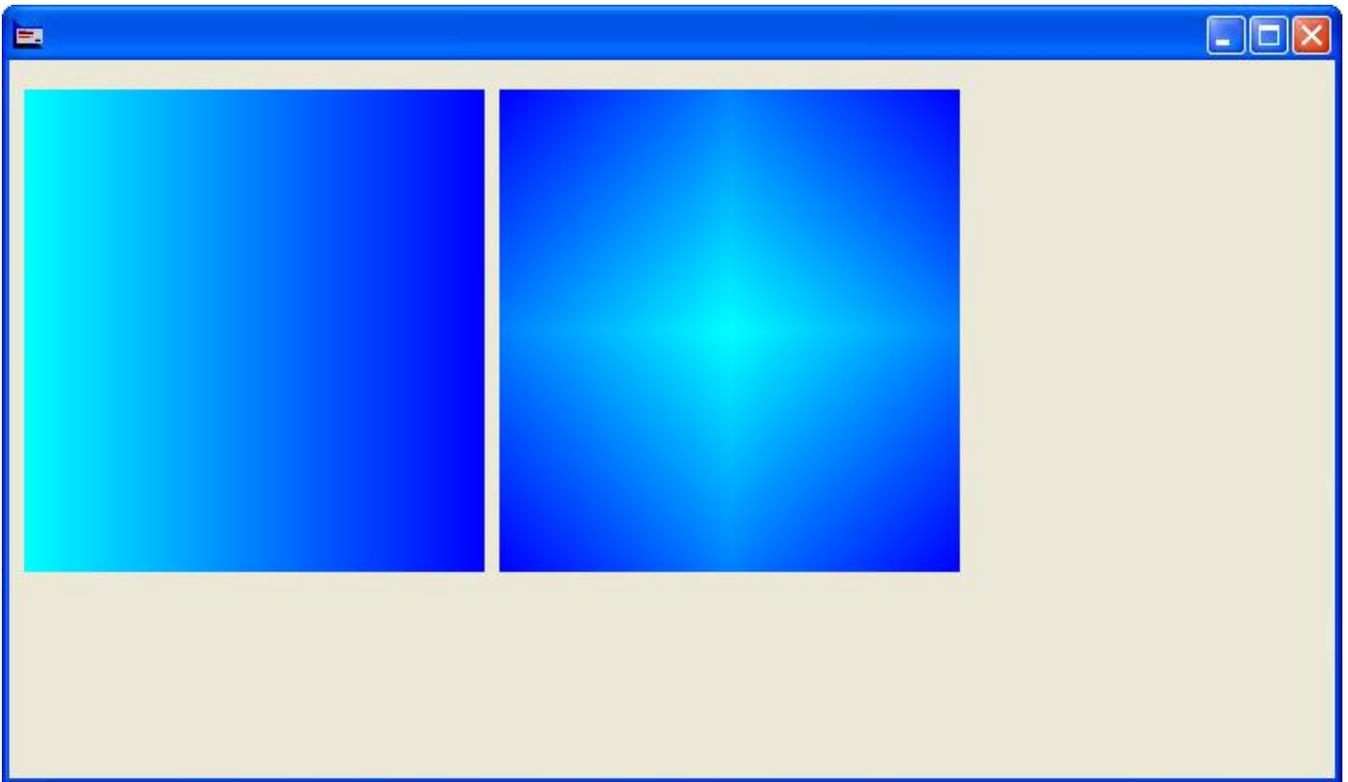
" Visible if neutralized " Option

If this option is checked, the line will be visible when neutralized. The line will also have the selected neutralization color.

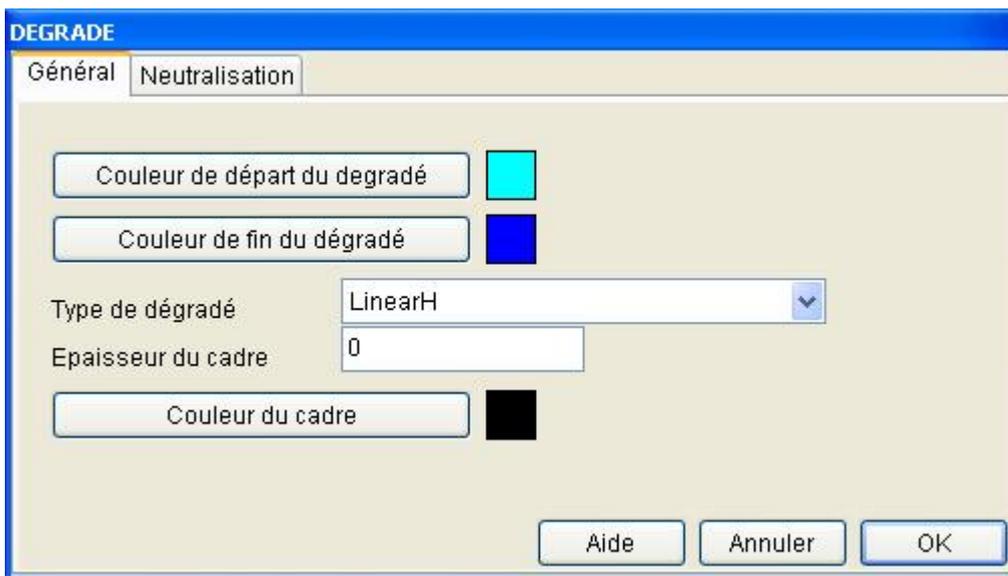
If this option is not checked, the line will not be visible when neutralized.

3.2.6.5.3 Gradient

This object draws a gradient on a supervisory form.



Dialog Box " Gradient" - General :



Begin color and end color for gradient :

This buttons allows to specify the colors used to draw the gradient.

Gradient type :

The user can specify the gradient type : circular, horizontal, vertical , and so one.

Frame thickness :

A frame can be drawn with the gradient. By default, there is no frame (Frame Thickness = 0)

Frame Color :

If the frame thickness is upper to 0, the frame color can be specified using this button.

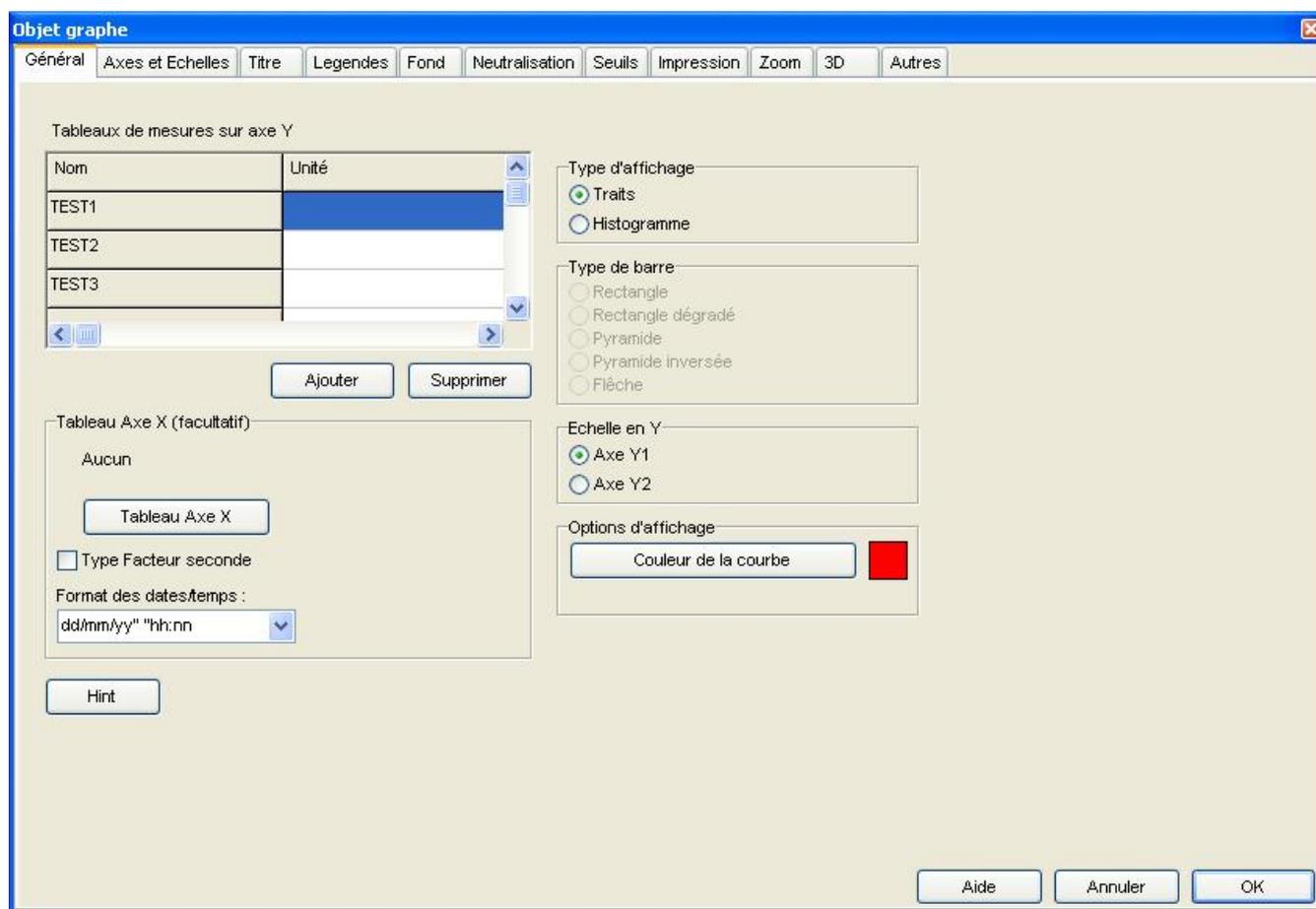
3.2.6.6 GRAPHIC tab objects

3.2.6.6.1 Graph

The object graph is used to display Curves or Histograms from valid samples tables coming from OPTIMA PLC or APIGRAF PLC. An Object graph can display 1 to 9 different curves.

In execution Mode, the curves can be printed. (See further: Printing)

Dialog box "Graph" - General



The parameters are :

"add " Button

Add a new curve to the graph. The add button opens an input box to enter the samples table name.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

"Delete" Button

Remove the selected samples table in the list if one table is selected.

X-Axe Table

This field is optional. The user can define a samples tab containing the X values. The other samples tables contains the Y values. If the X-Axe Table is defined, the values of this table will be used to set the X values corresponding to the Y values.

Otherwise if no X-Axe is defined, the Y values of the samples tables will be displayed at X=1,X=2,X=3,.....

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory

form with parameters ")

"Secondfactor type" option

This option is only used with APIGRAF samples tab containing samples with the "SecondFactor" format which is a specific APIGRAF format.

Date/Time Format

This option is only used if the parameter "Secondfactor" type is checked. In this case, the user can specify a format to display Date/Time of the X samples.

Examples of format :

dd / mm / yyyy" --- "hh:nn:ss displays 02/03/2002--06:23:15

AM / PM hh:nn displays 6:23PM

"d" dddd" "mm" "yyyy displays Saturday March 2002 2

Type of display

Permits to select the type of display for the curve selected in the list of the measure tables. There are two possibilities

- Milked
- Histogram

Bar type

This option is only accessible if the type of graph is a histogram. It permits to select the type of bar of the graph.

Scale in THERE

There are 2 available vertical scales in the graph, an on the left and (Y1 (Y2)) an on the right. This field permits to specify what vertical scale will be used for the display of the curve.

Color of the curve

This button permits to specify the color of the curve selected in the list of the measure tables.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

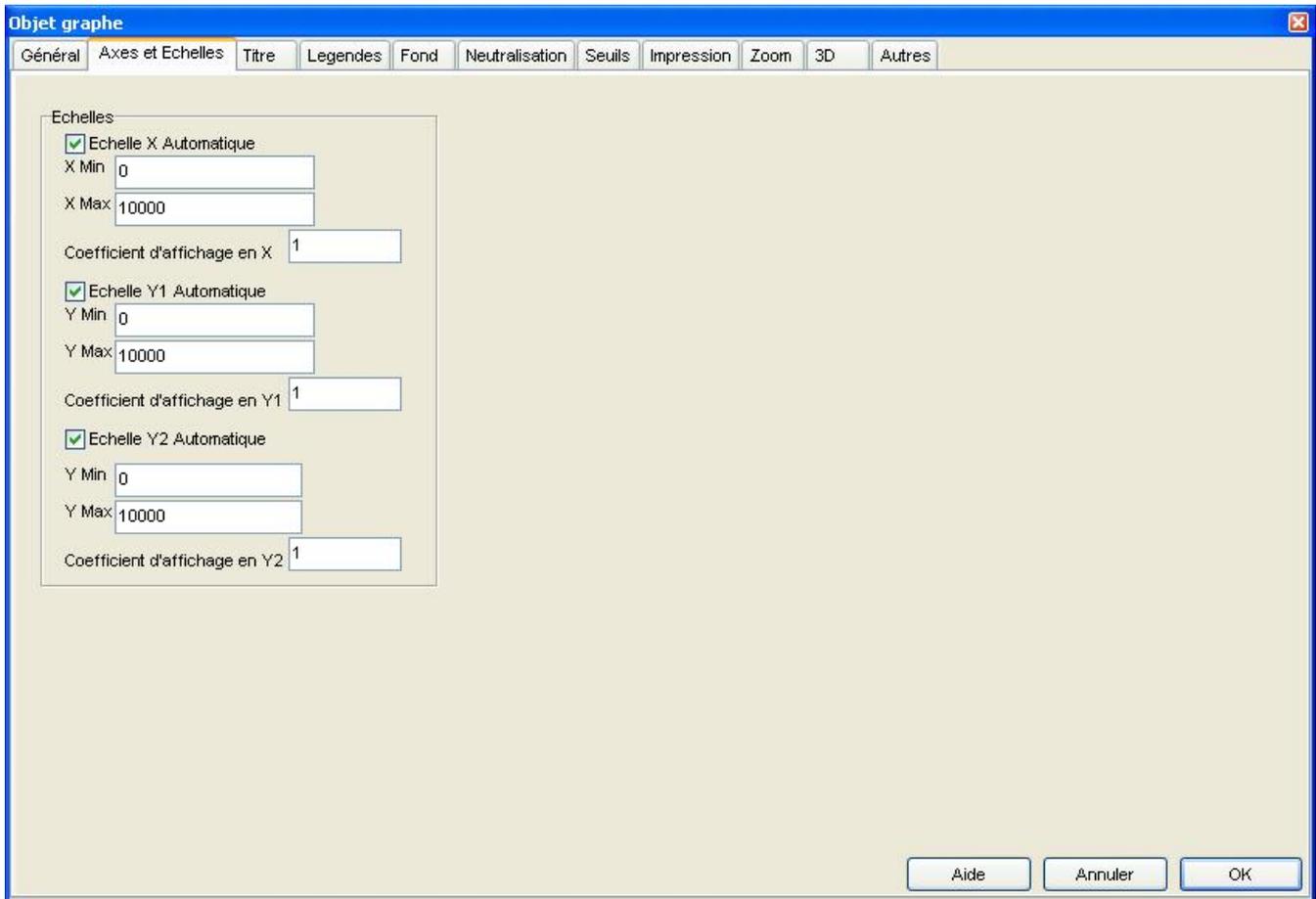
« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Dialog box graph - Tab "Axes and Scales"



Scales

Permits to specify the scales of the three axes following X, Y1 and Y2. The Y2 axis can be used in the case or several curves don't have the same units.

There are two fashions of scales, the automatic mode and the manual mode.

In the automatic mode, it is necessary to check the box " X scale, Y1 or Y2 Automatique " following the concerned axis. In this case, the values Min and Max of the scale will be adjusted to the values of the tables and the fields Min and corresponding Max to the scale don't have need to be informed.

If the box " Automatic scale " is not checked. The values min and max of the scale will be the values appearing in the fields Min and Max of the scale.

Coefficients of display

It is possible to specify for each of the three available scales a coefficient of display. This coefficient defines the report between the displayed value and the real value in the table.

Option "Scale X Automatique"

To check this option so that the extent of the scale of the X automatically adapts to the values of the curves. If this option is not checked, it is possible to give the XMin values and XMaxes of the scale of the X.

Coefficient of display in X

Its value is by default 1. This value is multiplied to the values of the measure table in X.

Option "Scale Y1 Automatique"

To check this option so that the extent of the scale of the Y1 automatically adapts (Scale of left) to the values of the curves. If this option is not checked, it is possible to give the Y1Min values and Y1Maxes of the Y1 scale.

Coefficient of display in Y1

Its value is by default 1. This value is multiplied to the values of the measure tables positioned in relation to the Y1 (axis of left) axis.

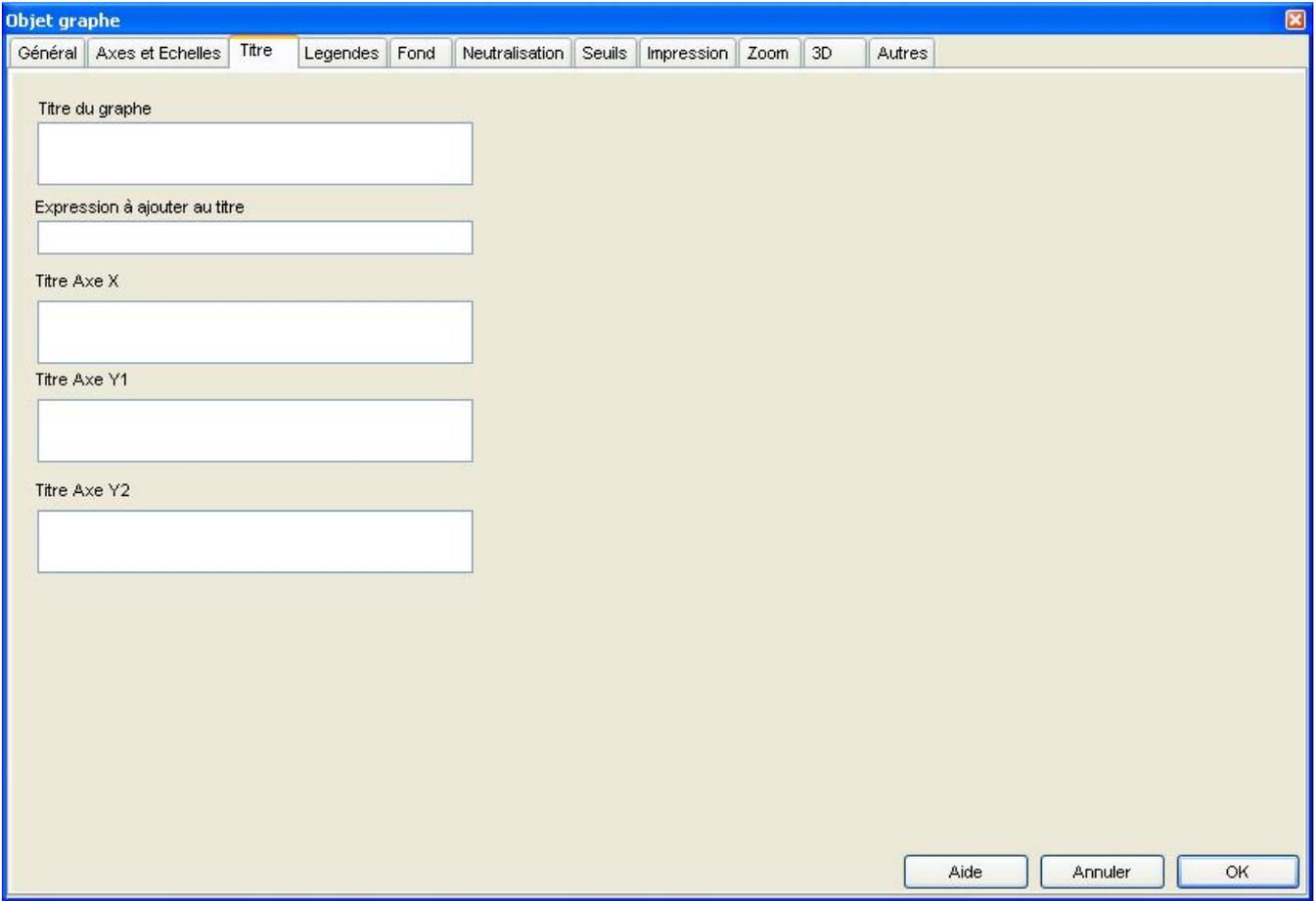
Option "Scale Y2 Automatique"

To check this option so that the extent of the scale of the Y2 automatically adapts (Scale of right) to the values of the curves. If this option is not checked, it is possible to give the Y2Min values and Y2Maxes of the Y2 scale.

Coefficient of display in THERE

Its value is by default 1. This value is multiplied to the values of the measure tables positioned in relation to the Y2 (axis of right) axis.

Dialog box graph - Tab "Titles"



The screenshot shows a software dialog box titled "Objet graphe" with a blue border. The "Titre" tab is selected, and the interface contains several text input fields for configuring graph titles. The fields are labeled as follows:

- Titre du graphe
- Expression à ajouter au titre
- Titre Axe X
- Titre Axe Y1
- Titre Axe Y2

At the bottom right of the dialog, there are three buttons: "Aide", "Annuler", and "OK".

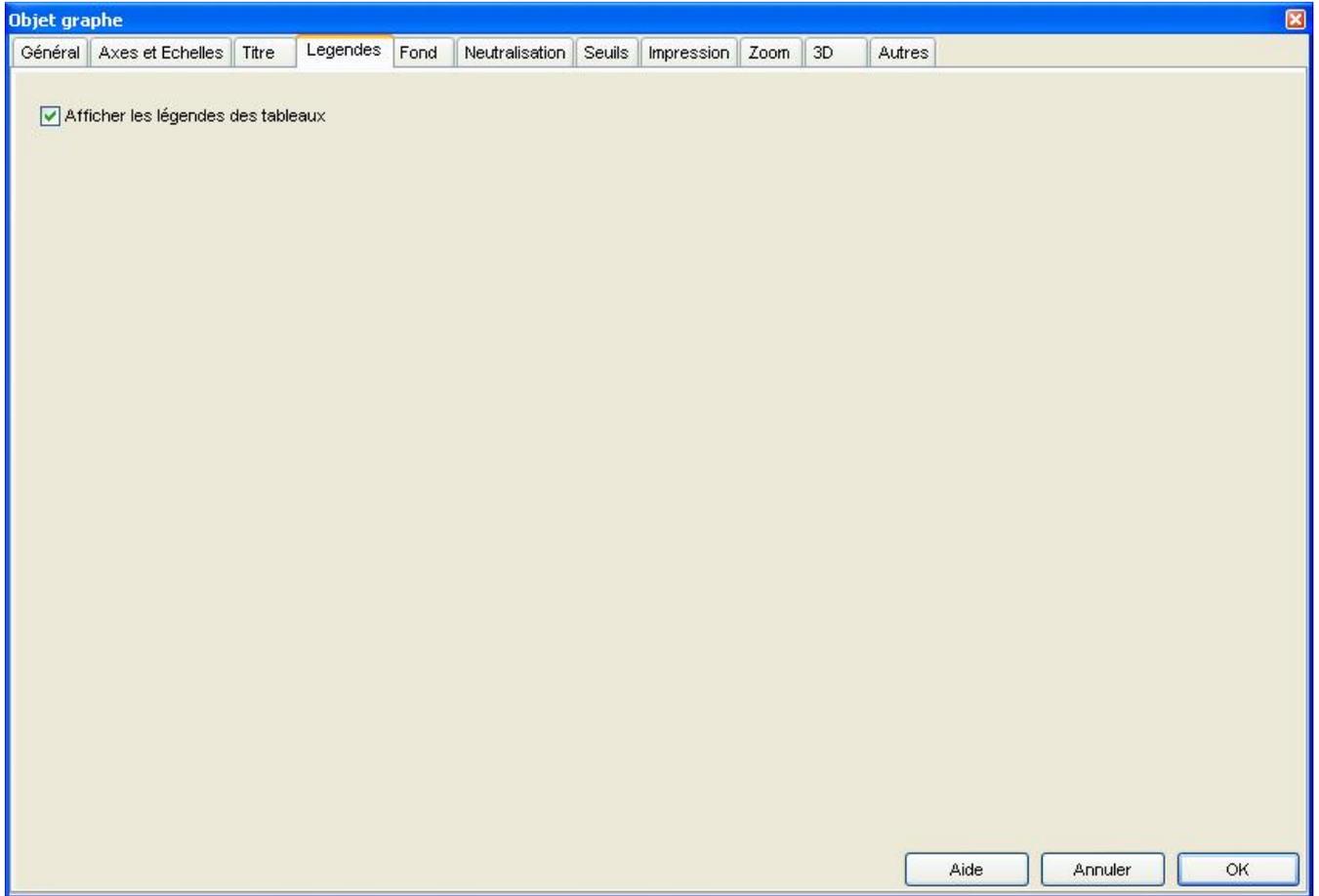
Titles

A general title can be given to the curve as well as legends for each of the three available axes.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

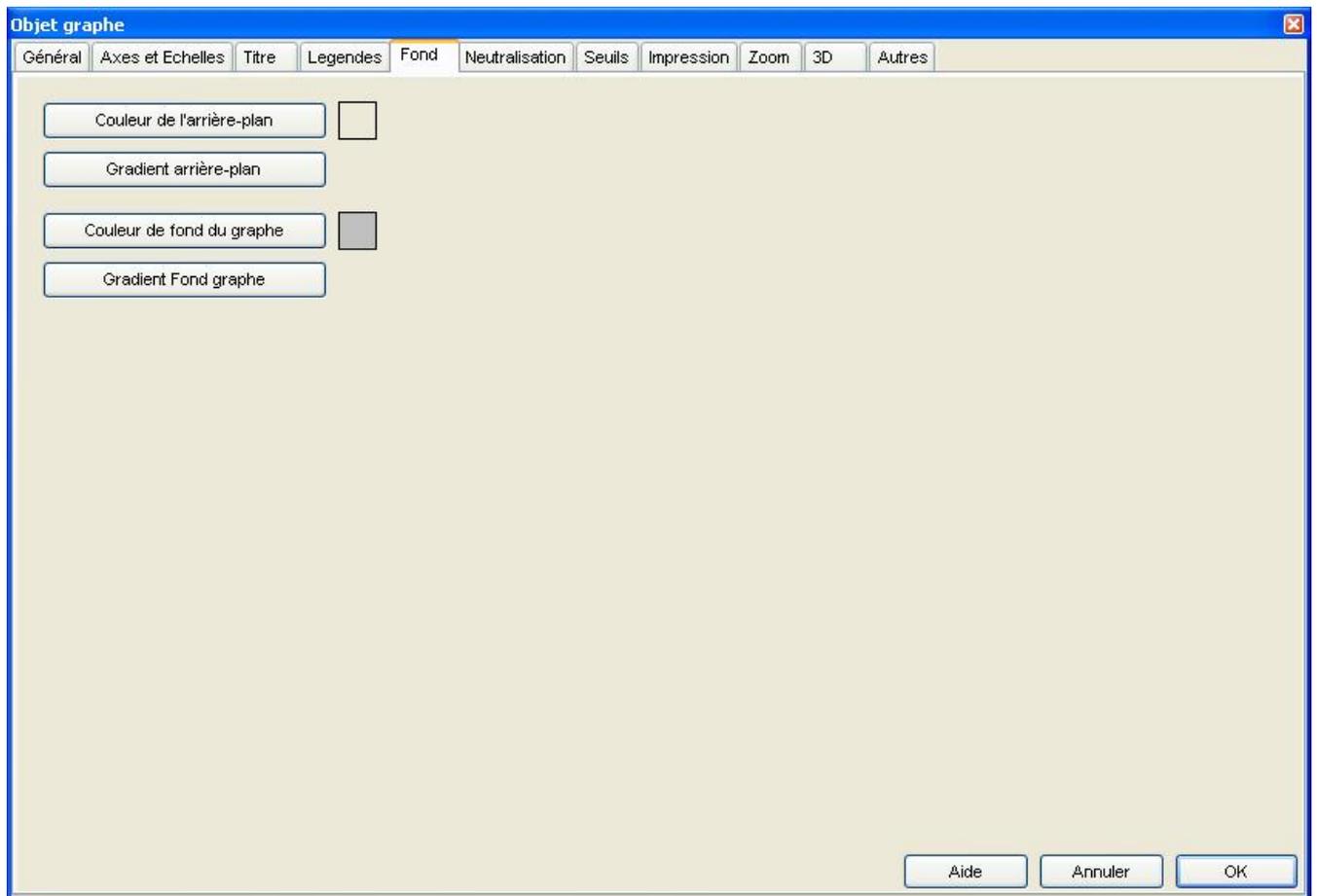
One can also specify an expression to add to the title. For example a serial number contained in a register: R100.

Dialog box graph - Tab "Titles"



To check the option to "Display the legends" to display the names of the samples tables associated to the colors of the corresponding curves.

Dialog box graph - Tab "Melts"



Color of the rectangle of the graph

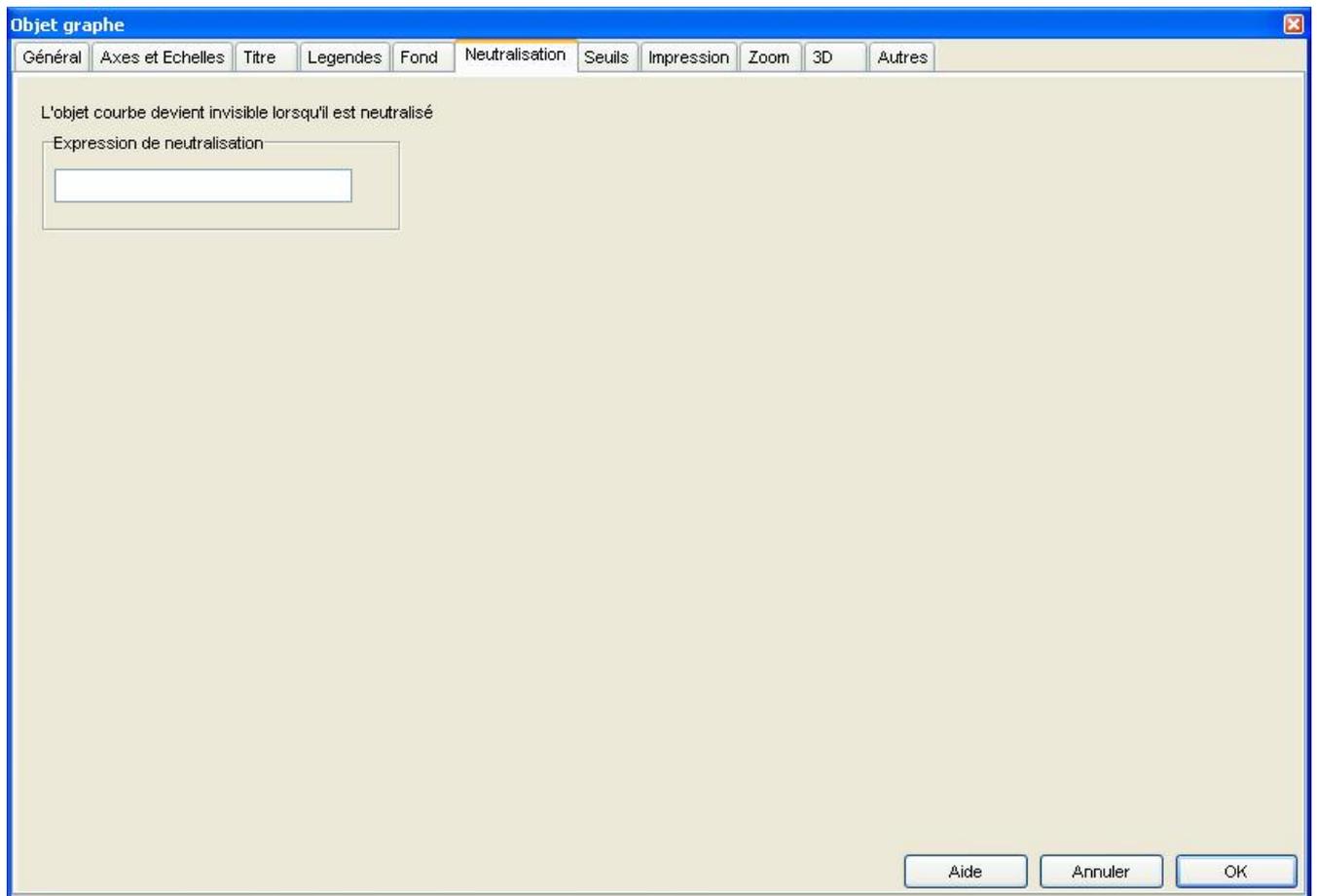
This button permits to choose the color of plane rear of the graph.

Color of the plane rear

This button permits to choose the color of the contour of the object graph.

Some buttons gradients also permit to define of damaged them of color in bottom of the object and behind plane of the curve.

Dialog box graph - Tab "Neutralization"



Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object bends will be invisible every time that the variable or the expression will be equal to true.

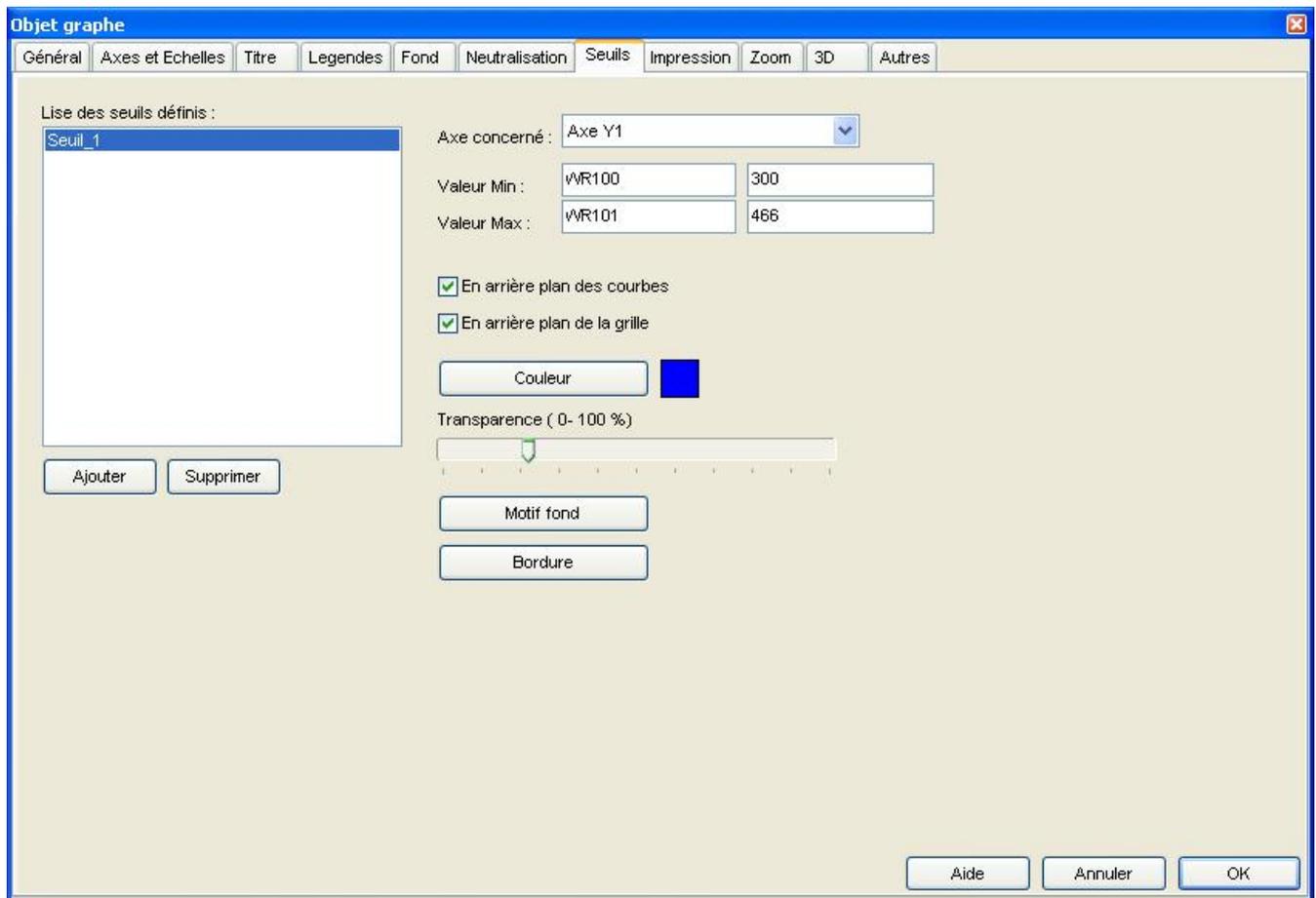
Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Dialog box graph - Tab "Doorsteps"



The doorsteps are the colored zones that show off on the curves and that permit to visualize some zones limits in general. Some doorsteps can be added or can be suppressed in the unwinding list with the help of the buttons "to Add" and "to Suppress." Once an added doorstep, it is possible to parametrize its aspect :

Axis concerned :

Defines the axis to which relates the doorstep, a doorstep can be therefore horizontal if it referred in Y1 or Y2 or vertical if it referred to the X axis.

Value Min and Value Max :

These values define the extent of the zone doorstep, one can enter the constant numeric values or the numeric expressions.

Example : R10

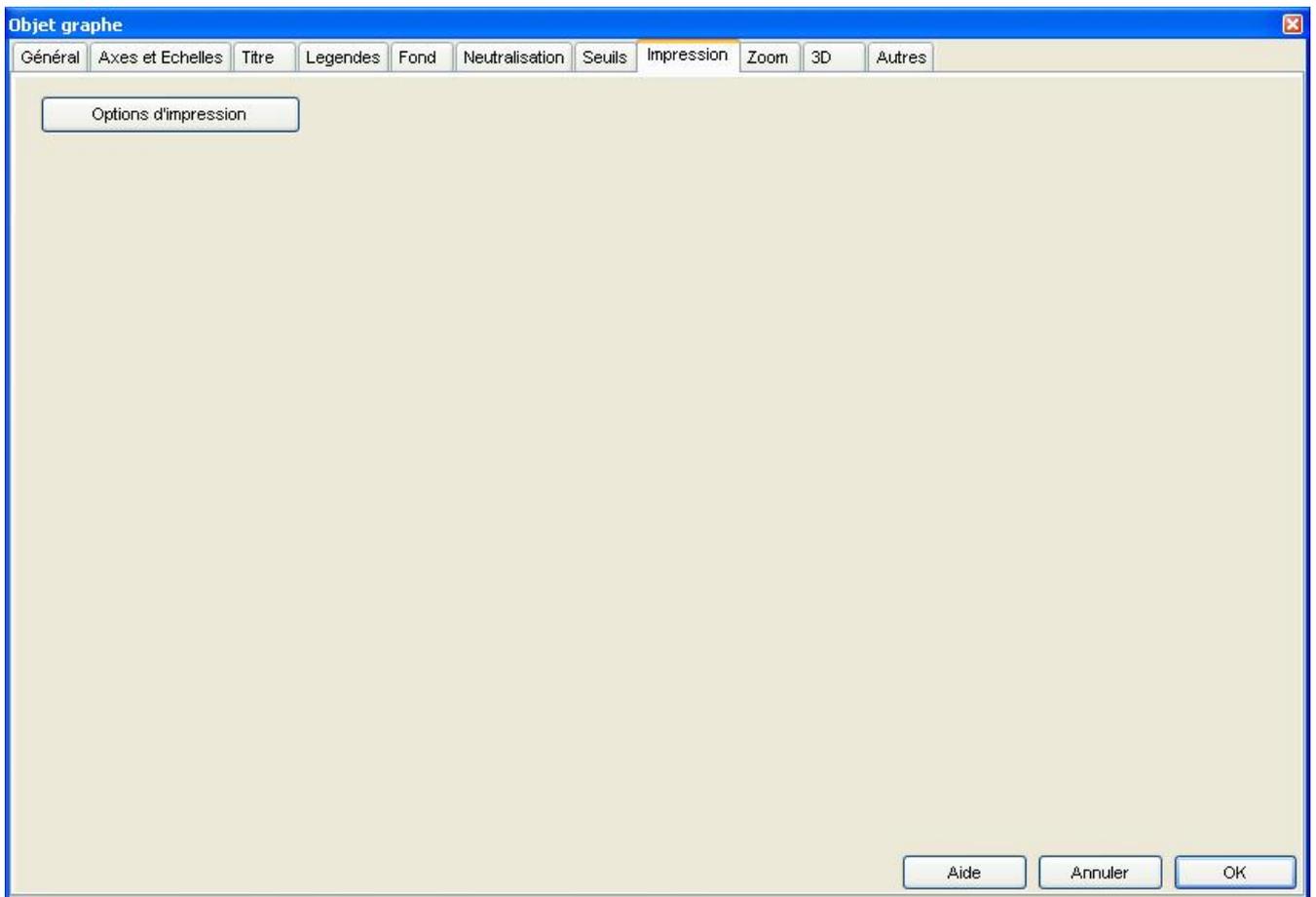
It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

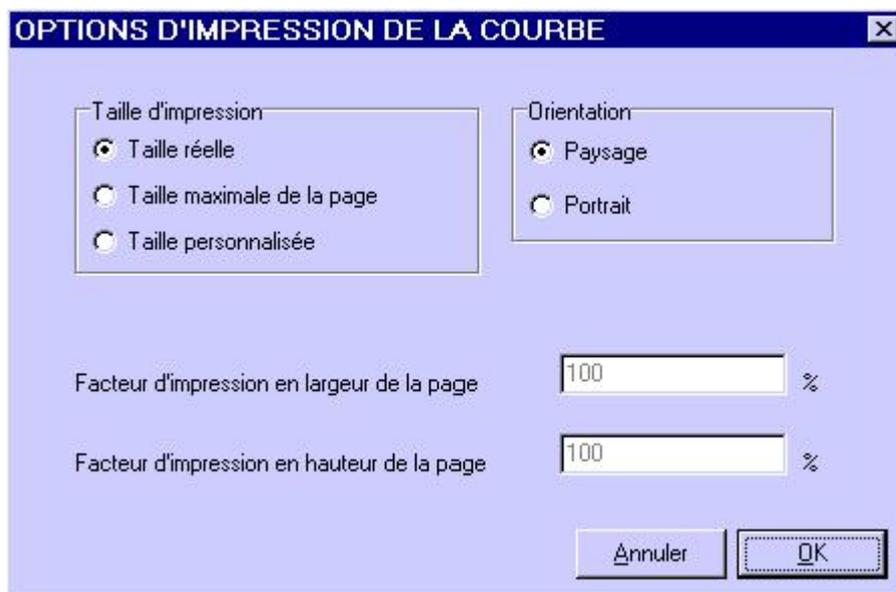
The appearance of the zone doorstep can also be redefined: the transparency, the motive of the bottom, the color,...

Dialog box graph - Tab "printing"



Options of printing

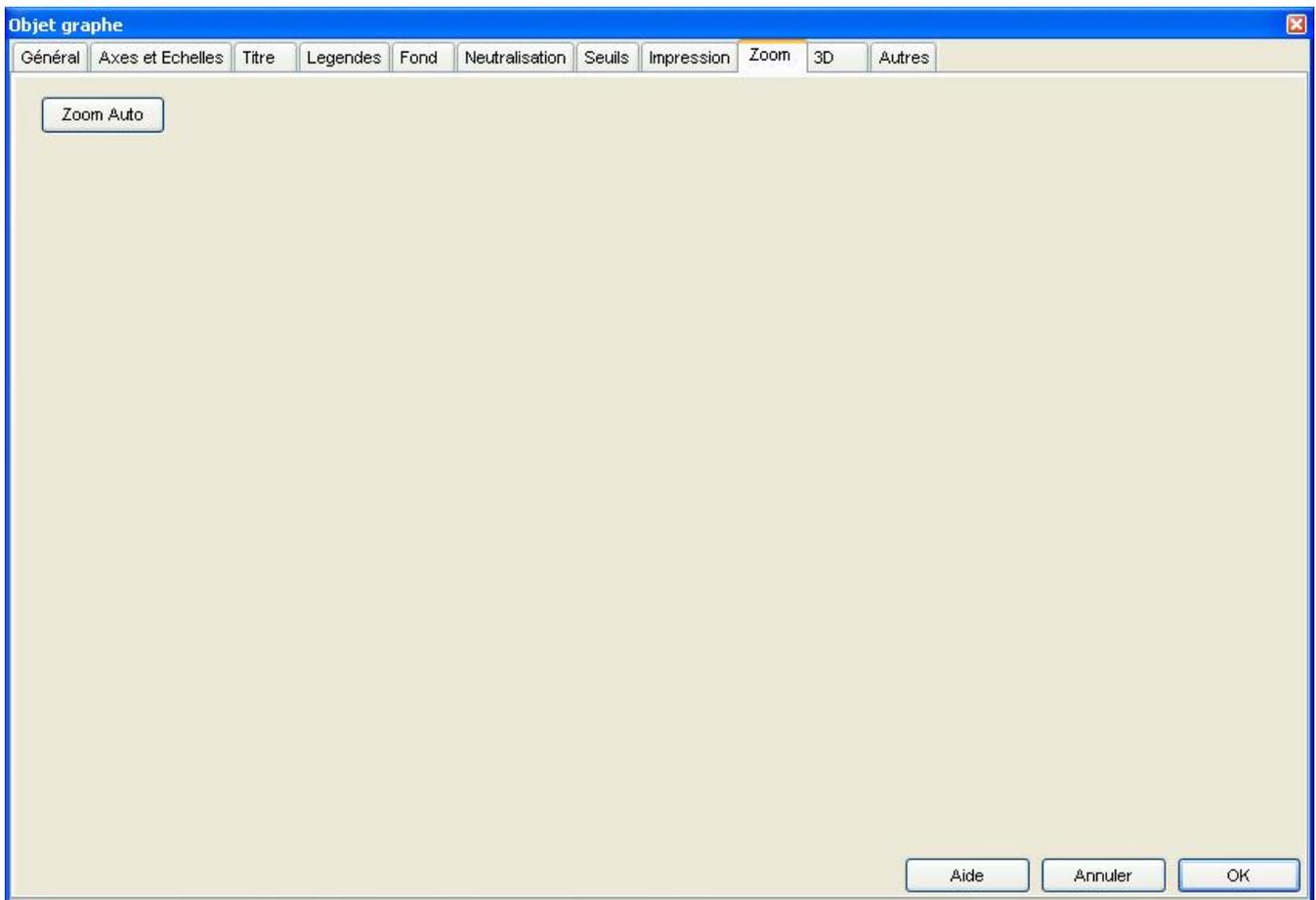
The curves can be printed at execution. It is necessary for it to double-click on the curve to print at execution. The options of printing permit to parameterize the printing of the curve :



If the option " Size personalized " is chosen, the factors of printing in width and in height of the page will be taken in account for the printing. These factors will be applied to the size of display to the screen of the curve.

The option " Maximal size of the page " go spread the curve on the whole size of the output paper. In this case the definition of printing is less good.

Dialog box graph - Tab "Zoom"



The button "Zoom" opens the following window :

Zoom Automatique

Expression zoom plus abscisses :

Expression zoom moins abscisses :

Pas du zoom en pixels :

Expression shift plus abscisses :

Expression shift moins abscisses :

Pas du shift en pixels :

Expression reset zoom :

Annuler OK

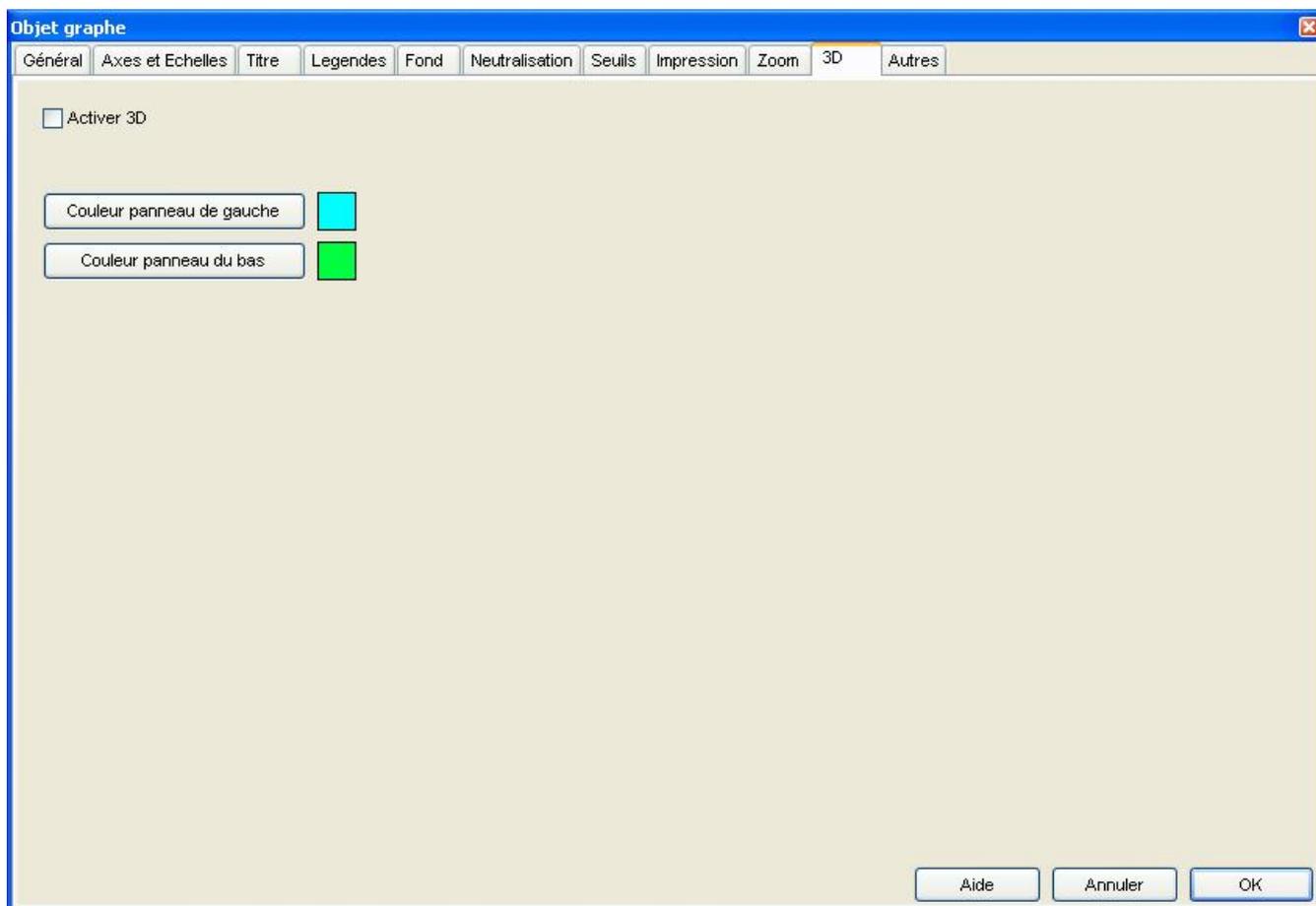
This window permits to associate variables of control of the zooms of the curve.

Example : To enter SA10 in the input box "Expression zoom more abscissas." To define a button then API monostable with the SA10 value. To every pressure of the button, the curve is going to be zoomed in in X with the specified (2 by default) step.

One can make the same thing to control the displacement ("Shift") of the curve.

An expression reset zoom permits to annul all zooms done on the curve.

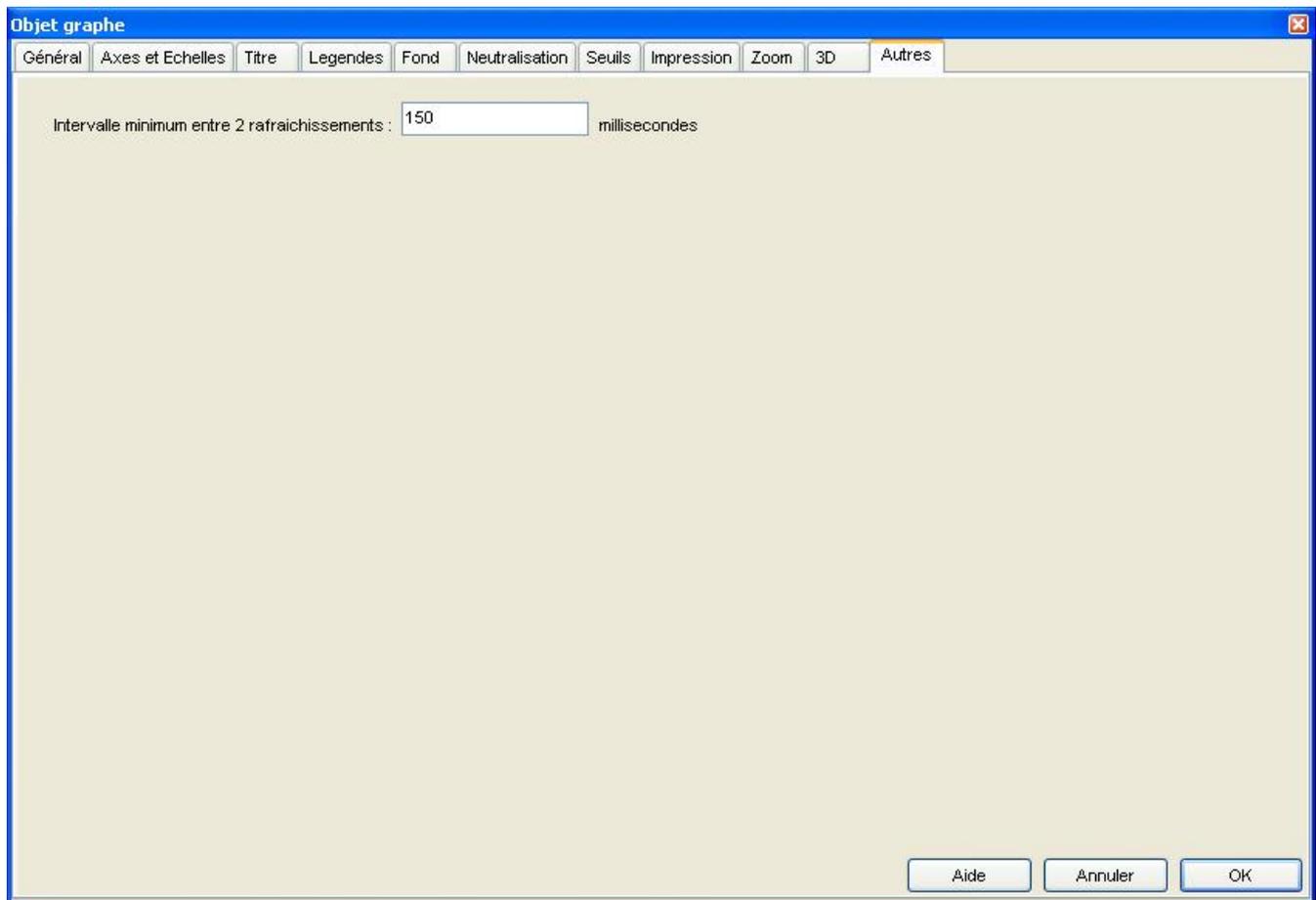
Dialog box graph - Tab "3D"



Option to "Activate 3D"

Permits to display the window of the graph in 3 measurements. One can define the colors of the panels of left and the low created by the visualization 3D then.

Dialog box graph - Tab "Others"



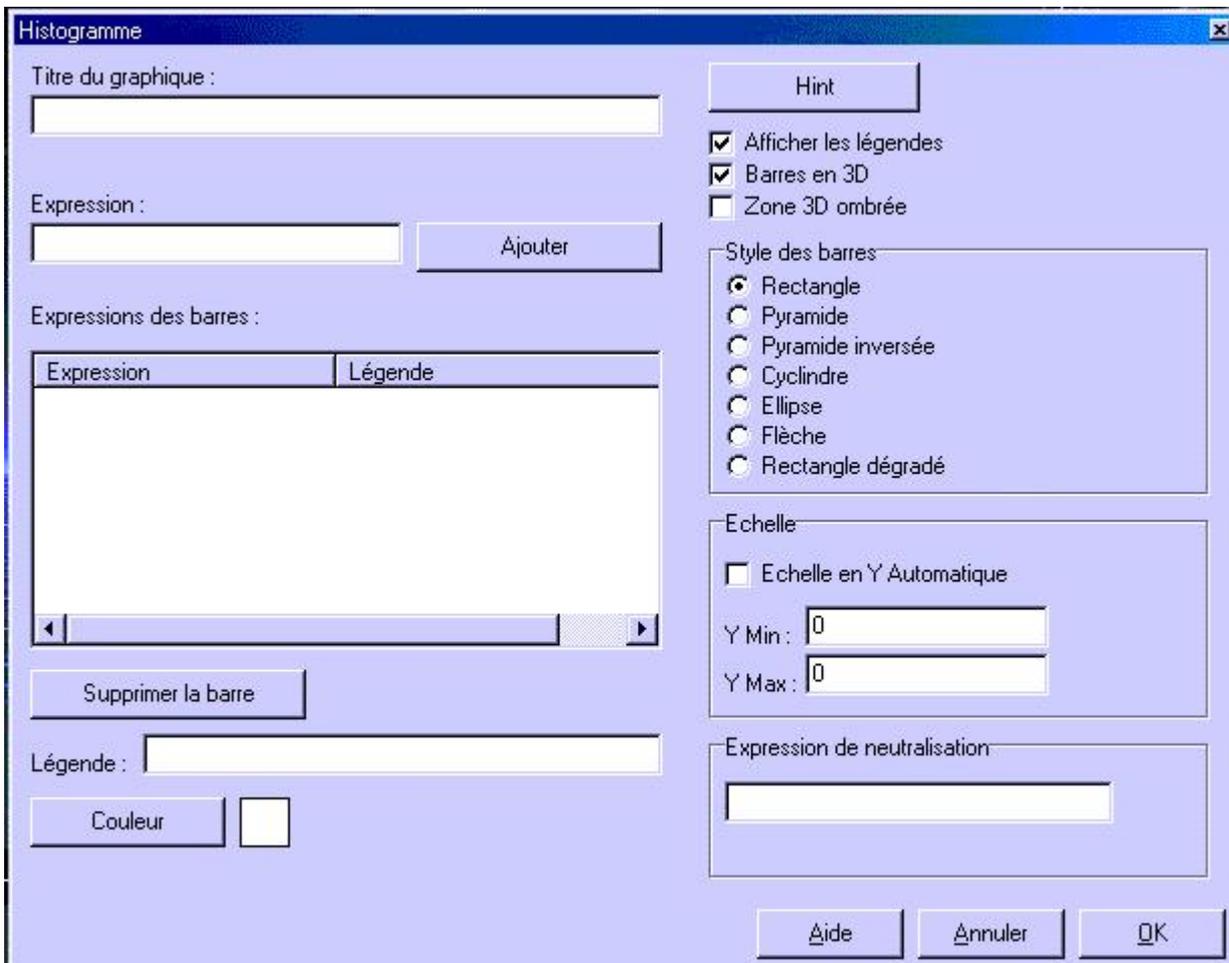
The complex displays of voluminous samples tables need a lot of resources. The user can limit the refresh rate of the displays.

3.2.6.6.2 Histogram

The object histogram permits to display a graph of type histogram on a supervisory form. The user must define the list of the variable PLCs that is going to compose the different bars of the histogram.

These variables must be of type numeric integer or decimal (Examples: R100 or RR100)

Dialog box " Histogram " :



The fields are the next one :

Title of the diagram

Permits to specify the title that will appear in top of the diagram. It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Expression

This window can contain the name of a variable PLC of numeric type. This variable can be integer or decimal. Once the input done, it is necessary to click on the button " To add " to add the expression in the list of the bars.

Example : R10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Button " To add "

Add the expression seized in the window " Expression " in the list of the bars

Button " To suppress the bar "

To suppress a bar, it is necessary to select the corresponding expression first in the list and then to click on the button " To suppress the bar"

Input window "Legend"

Permits to modify the legend that will be associated to a bar. It is necessary to select the expression of the corresponding bar first in the list to modify its legend then.

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Button " Color "

Permits to modify the color of a bar. It is necessary to select the bar first in the list to modify its color then.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Option " To display the legends "

If this option is not checked, the containing setting the list of the legends won't be displayed and the histogram will occupy this fact a more important place in the object.

Option " Bar in 3D "

If this option is checked, the bars composing the histogram will be drawn in 3 measurements.

Option " Zone 3D Shaded "

If this option is checked, the part 3D of the bars will appear darker than the sector.

Style of the bars

Permits to choose the appearance of the bars forming the histogram (Rectangle, arrow.)

Scale in THERE Automatic

If this option is checked, the values min and max of the vertical scale will automatically be calculated.

Ymin

If the option " Scale in THERE Automatic " is not checked, this value will fix the value min of the vertical scale.

YMax

If the option " Scale in THERE Automatic " is not checked, this value will fix the value max of the vertical scale.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object histogram will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.6.3 Scale

This object permits to display a scale stepped up on a supervisory form, this object can be useful to add a graduating under an object Sizes up, cursor etc.

Example of object scale :



Dialog box " Scale " :

The parameters are the next one :

Minimal value

This parameter indicates the initial value of graduating of the scale. This parameter can be integer or decimal.

Maximal value

Indicate the final value of graduating of the scale. This parameter can be integer or decimal.

Interval

This parameter indicates the value separating two graduating. This parameter can be integer or decimal.

Number of decimals

The graduating can be displayed with decimals. This parameter indicates the number of decimals displayed. This parameter is of integer type.

Prefix

Permits to define a text destined to be added on the left of the numeric values of graduating.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Suffix

Permits to define a text destined to be added on the right of the numeric values of graduating (for example, the

suffix can be used to display a% on the right of every value).

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Button " Color of the scale "

This control permits to modify the color of the scale.

Button " Characters "

Permits to define the font used to display the numeric values of the graduating.

Orientation

Permits to orient the graduating in a vertical or horizontal way.

Position

If the graduating is horizontal, this control permits to place the numeric values of graduating in top or at the bottom of the scale.

If the graduating is vertical, this control permits to place the numeric values of graduating on the left or on the right of the scale.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object scale will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

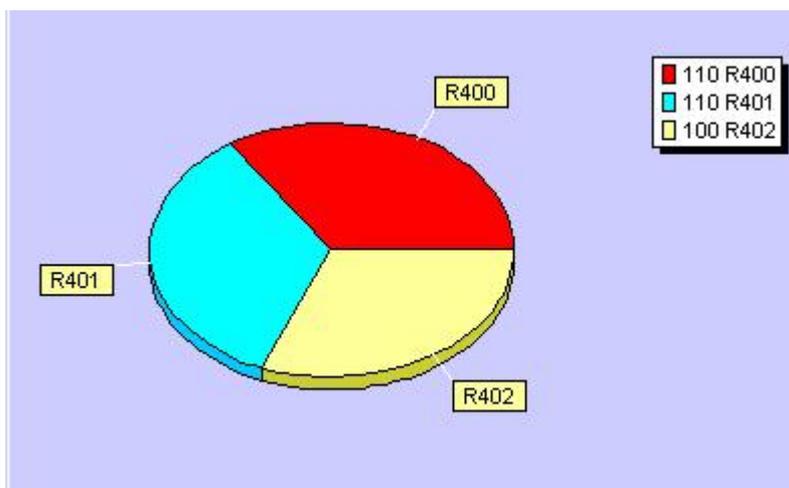
3.2.6.6.4 Pie graph

The object pie chart permits to display a graph of type pie chart on a supervisory form. The user must define the

list of the variable PLCs that is going to compose the different sectors of the pie chart.

These variables must be of type numeric integer or decimal (Examples: R100 or RR100)

Example of pie chart :



Dialog box " Camembert " :

Expression	Légende
R400	R400
R401	R401
R402	R402

The fields are the next one :

Title of the diagram

Permits to specify the title that will appear in top of the diagram. It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Expression of the sector

This window can contain the name of a variable PLC of numeric type. This variable can be integer or decimal. Once the input done seizure, it is necessary to click on the button " To add " to add the expression in the list of the sectors.

Example : R10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Button " To add "

Add the expression seized in al window " Expression of the sector " in the list of the sectors.

Button " To suppress "

To suppress a sector, it is first necessary to select it in the list and then to click on the button " To suppress "

Window of seizure " Legend "

Permits to modify the legend that will be associated to a sector. It is necessary to select the sector first in the list to modify its legend then.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Button " Color "

Permits to modify the color of a sector. It is necessary to select the sector first in the list to modify its color then.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Option " To display the legends "

If this option is not checked, the containing setting the list of the legends won't be displayed and the pie chart will occupy this fact a more important place in the object.

Option " Zone 3D Shaded "

If this option is checked, the part 3D of the area (slice of the sector) will appear darker than the sector.

Option " Circle perfects "

If this option is checked, the pie chart will form a circle perfects. Otherwise the shape of the pie chart will be elliptic.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object pie chart will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

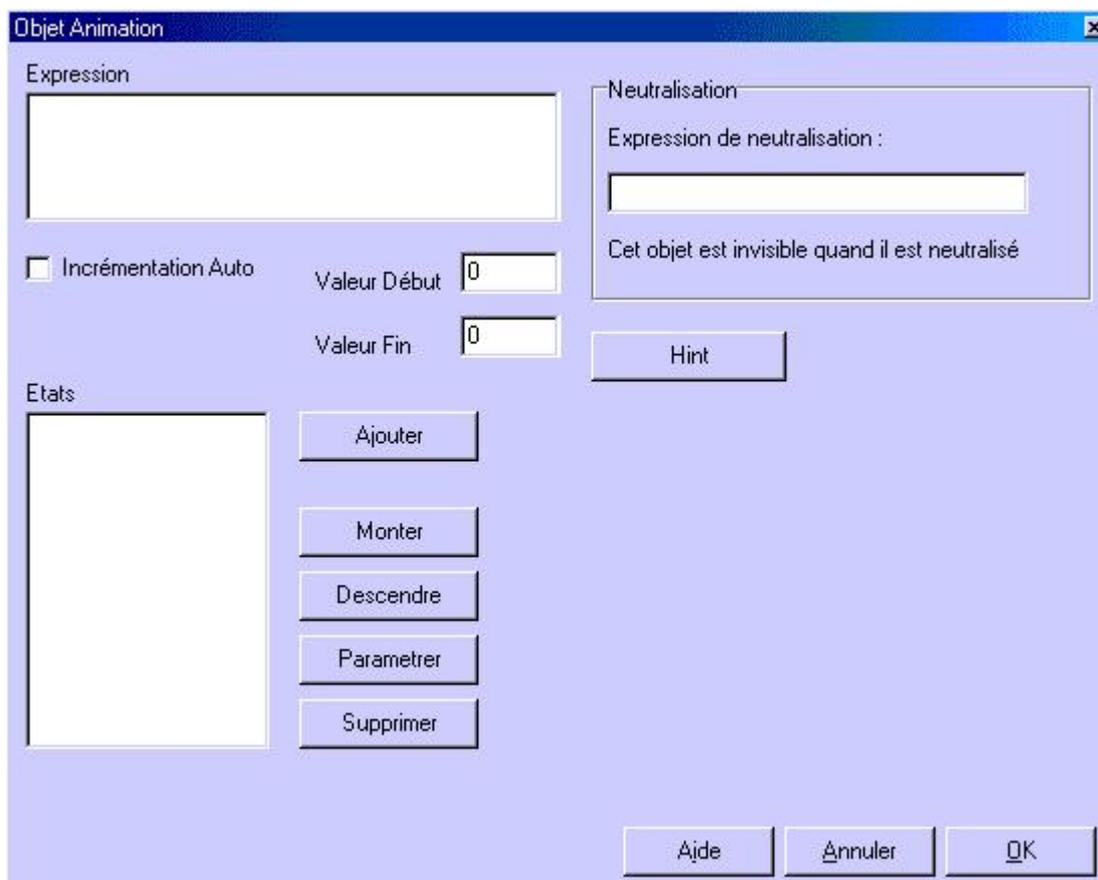
3.2.6.7 ANIMATIONS / PICTURES tab objects

3.2.6.7.1 Animation

This object is generally used to achieve an animation, of the buttons to multiple states, or to display contextuels. The view of this object change according to the value of its control variable.

This variable can be of boolean type (L flag, logical Switch SA, ...) or integer (Register R.).

Dialog box Object Animation



The parameters are the next one :

Expression

Permits to define the name of the control variable (variable associated to the object animation). This variable acts as an index for every state of the animation (a state corresponds to a value of the variable).

To indicate in this field the variable PLC to use (of integer numeric type)

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R20

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Beginning and Value end

These fields permit to define the range of values of the control variable, which will define the different aspects of the object. For example, if the variable of control is of boolean type the value of beginning will be 0 and the one of end will be 1, we will get an object therefore in two states as, for example, a button button bistable.

You can also define a first state for value range 0-10, another state for range 11-13, another state for value 14,...

Increment Auto

If the box " Auto Increment " is checked, the variable is incremented automatically each time a mouse click event is done on the object animation. When the end value is reached, the begin value is set in the control variable.

States of the variable

This zone permits to visualize the list of the different states of the variable to test and the corresponding display. This list is displayed with the value of beginning and the value of end for every state.

To add, to Suppress, to Parameterize

The button " To add " permits to create a new state. The button " To suppress " suppress the state in surbrillance in the list. Finally, the button " To parameterize " permits to modify the state in surbrillance (a double click out of one of the states provokes the same effect), it allows the call the dialog box " State of animation " (cf. below).

To bring up, to Descend

These buttons permit to baffle the bar in surbrillance in the list: upwards with " to go up " and downwards with " to descend ".

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object animation will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

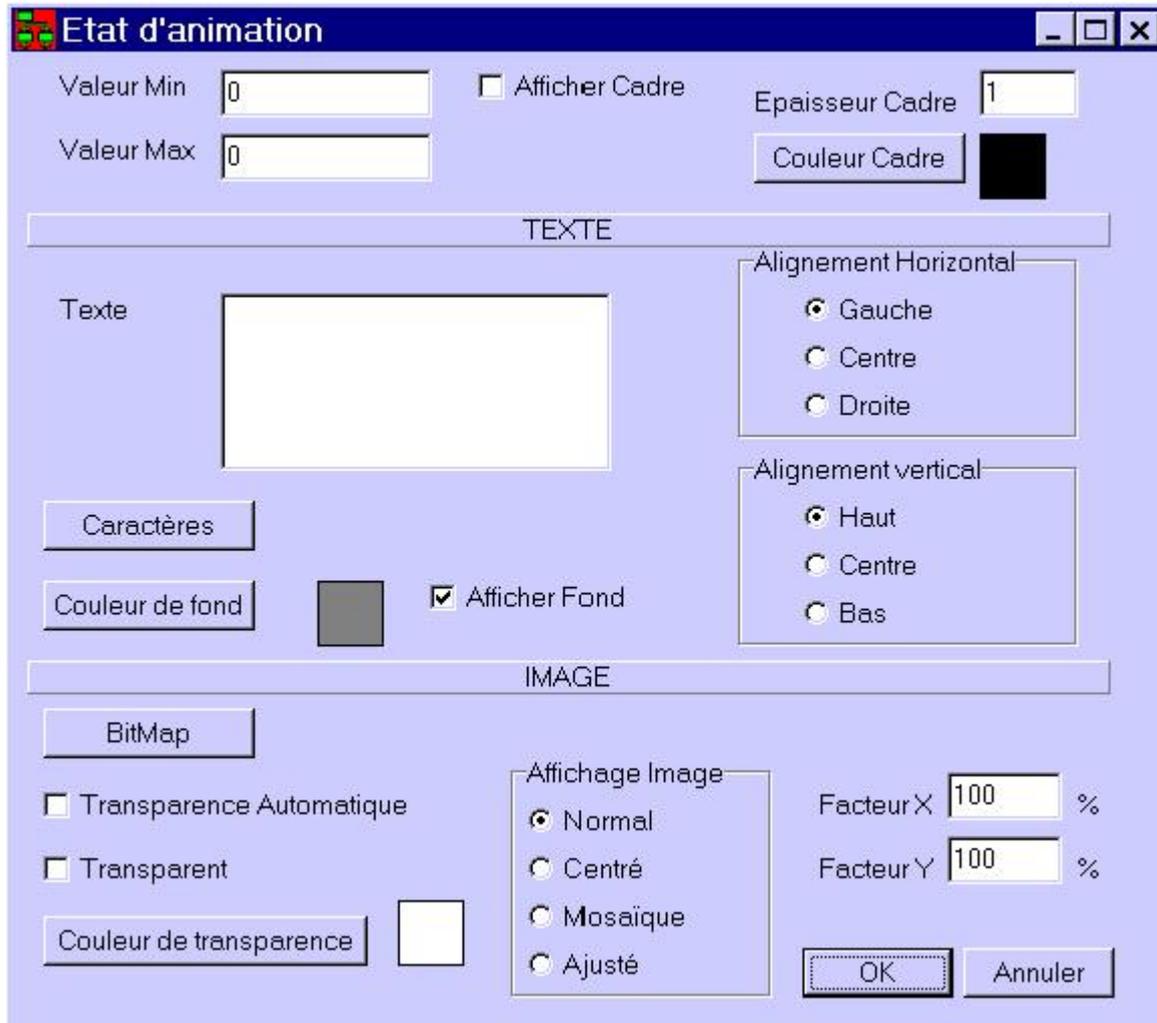
« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Box of animation State dialog



This dialog box permits to configure a display given for a value of the variable associated.
The following fields owe definite beings :

Value min and value max

These two fields permit to define the first and the last value of the variable for which the display will be achieved with the parameters that follow.
In other words, these values define the interval of values, included boundary-marks, corresponding to the state of the object describes in this box.

To display setting

This option permits to display a setting around the picture of the object.

Color centers

Permits to choose the color of the setting with the help of the dialog box color.

Thickness centers

This option permits to choose the thickness of the setting of the picture. The value is by default 1 pixel.

Text

To enter in the corresponding zone the text to display.
It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Groups vertical Alignment and horizontal Alignment

Permits to define the manner whose text must appear in the object.

Characters

Allows the access the dialog box concerning the type of display of the text (color, police.).

Color of bottom

Selection of the color of bottom of the object by means of a dialog box " color ".

To display bottom

If this option is not checked, the object is transparent, otherwise the bottom takes the higher definite color.

Bitmap

Permits the import of a format bitmap (BMP) picture. To import the picture it is sufficient to indicate the path of access to the file. The button " to browse " facilitate research by displaying the list of the BMP files.

The display of the picture can be also

- Normal: the picture is displayed on the left in top of the object,
- Centered: the picture is displayed in the center of the object,
- Mosaic: the picture is repeated in order to cover the whole surface with the object.
- Adjusted: permits to adjust the size of the picture to the one of the object.

Automatic transparency

If this option is checked, the color of transparency is automatically determined. It is in principle the color of bottom of the picture that is taken in account.

Transparent

Permits to define if one wants that the picture has a color of transparency. So such is the case, the color of the picture defined like color of transparency won't be displayed. The picture lets appear what is behind plane to the level of the transparency color then.

Color of transparency

This option permits in the case or the option " Automatic transparency color " is not checked to choose the color of transparency of the picture.

X factors and THERE

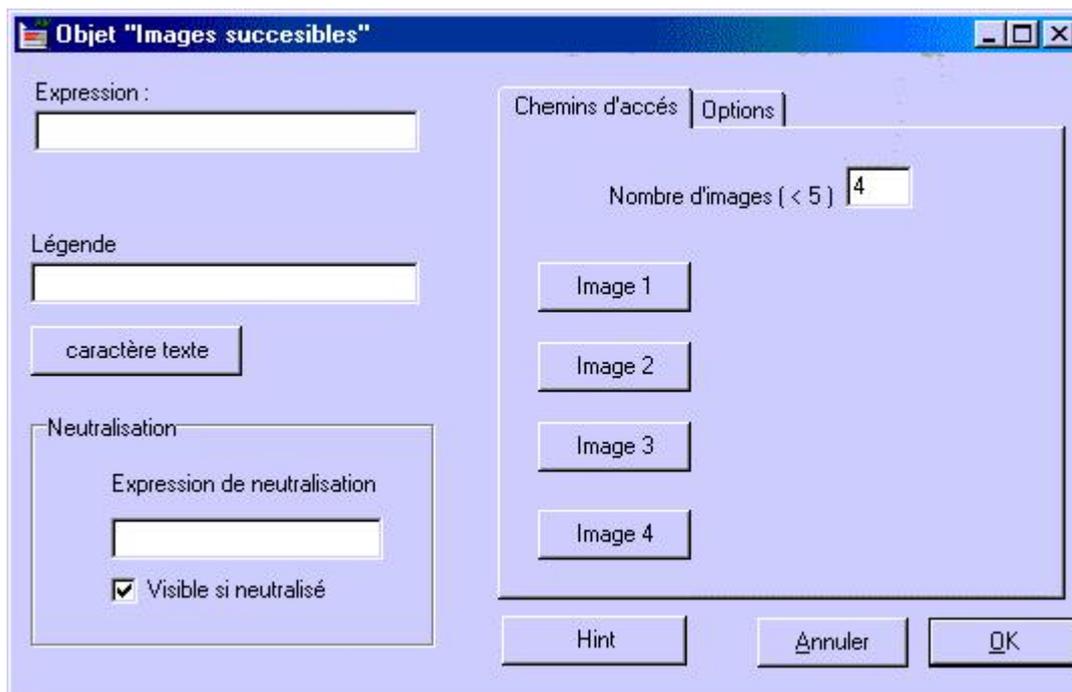
Permits to modify the size of the picture following two scales (vertical and horizontal) capable to vary from 0 to 1000%.

3.2.6.7.2 Pictures animated

This object is generally used to achieve an animation, of the buttons to multiple states, or to display contextual messages.

The appearance of this object varies with a click on the left button of the mouse above the object or according to the value of its control variable. This variable can be of boolean type (L flag, logical Switch SA) or integer (Register R).

Dialog box " Insertion successive pictures " :



The parameters are the next one :

Expression

Permits to define the name of the variable of control of the object (variable associated to the object insertion successive pictures). This variable acts as index for every state of the animation (a state corresponds to a value of the variable). To indicate in this field the variable PLC to use (of integer numeric type)

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R20

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text "

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the object will become invisible when it will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Tab " Paths of access "

The parameters are the next one :

Number of pictures

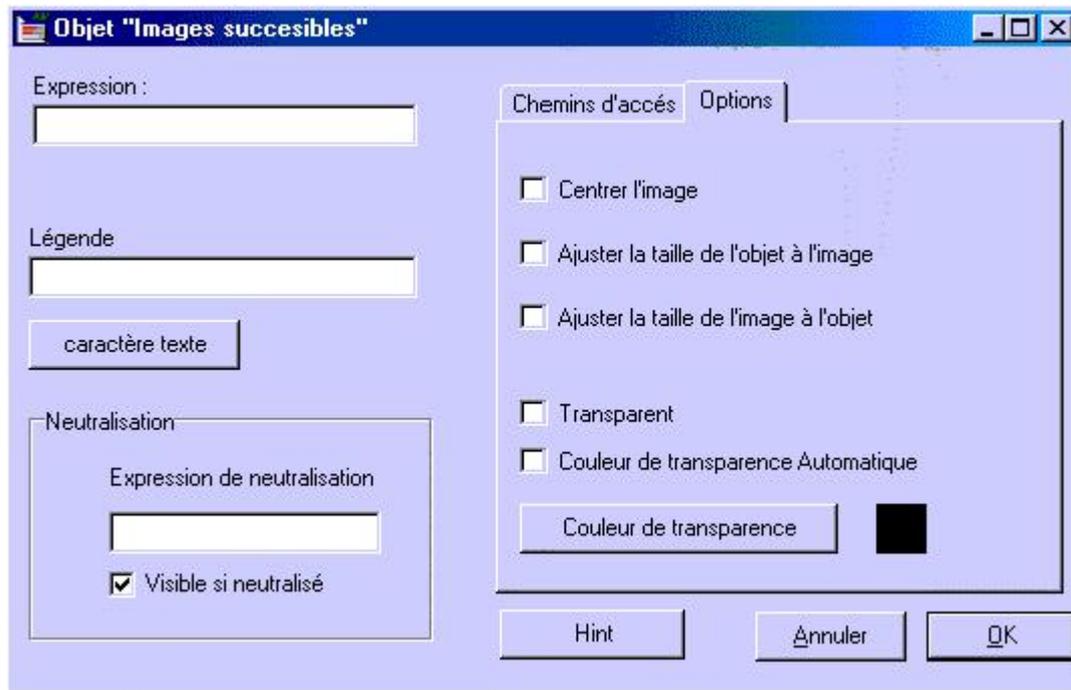
Permits to specify the number of picture that will follow each other during the animation. This parameter must be an encode understood between 1 and 4.

Picture 1, Picture 2, Picture 3, Picture 4

Permits to specify the corresponding file to the picture. The following formats are admitted: * .BMP; * .EMF; * .WMF; * .ICO

Remark : A collection of pictures to the format BITMAP is available in the APIGRAFIP\GRAPHIQUES\BITMAPS subdirectory

Tab " Options "



The parameters are the next one :

Option to Center the picture

This option permits to center the picture in the object.

Option to Adjust the size of the object to the picture

This option automatically adjusts the size of the object creates to the real size of the picture.

Option to Adjust the size of the picture to the object

This option automatically adjusts the size of the picture to the real size of the object. The fact of redimensionner the object will have an influence therefore on the size of the picture.

Transparent

Permits to define if one wants that the picture has a color of transparency. So such is the case, the color of the picture defined like color of transparency won't be displayed. The picture lets appear what is behind plane to the level of the transparency color then.

Automatic transparency color

If this option is checked, the color of transparency is automatically determined. It is in principle the color of bottom of the picture that is taken in account.

Color of transparency

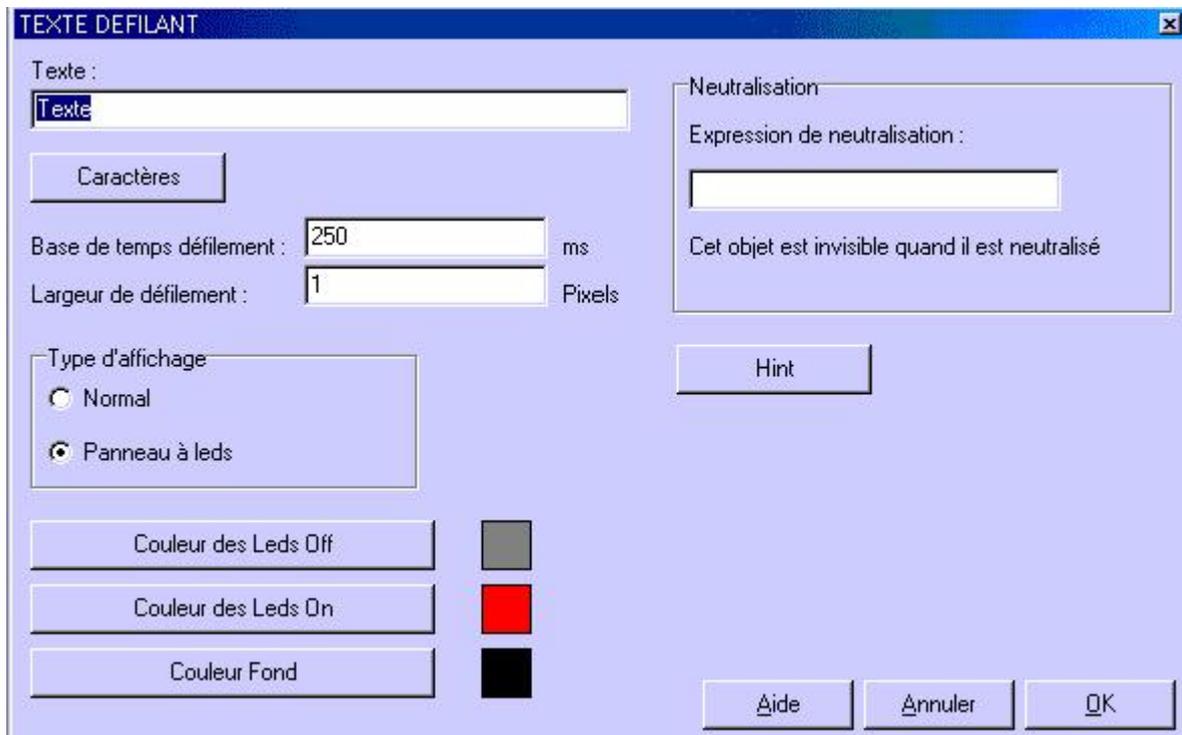
This option permits in the case or the option " Automatic transparency color " is not checked to choose the color of transparency of the picture.

3.2.6.7.3 Text parading

This object permits to display a text that parades in a continuous way. Two aspects are possible for this object :

- the aspect normal text
- the aspect Panel to leds

Dialog box text parading



The parameters are :

" Text "

Defines the text to make parade.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Button characters "

This button permits to define the police of the text.

" Type of display "

2 types of display are available :

- normal
- panel to leds

Example of display with the type panel to leds :



" Button Color of the leds off "

Defines the color of the leds that is extinguished.

" Button Color of the leds one "

Defines the color of the leds lit.

" Button Color Melts "

Defines the color of bottom of the zone of text scrolling.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object Text parading will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

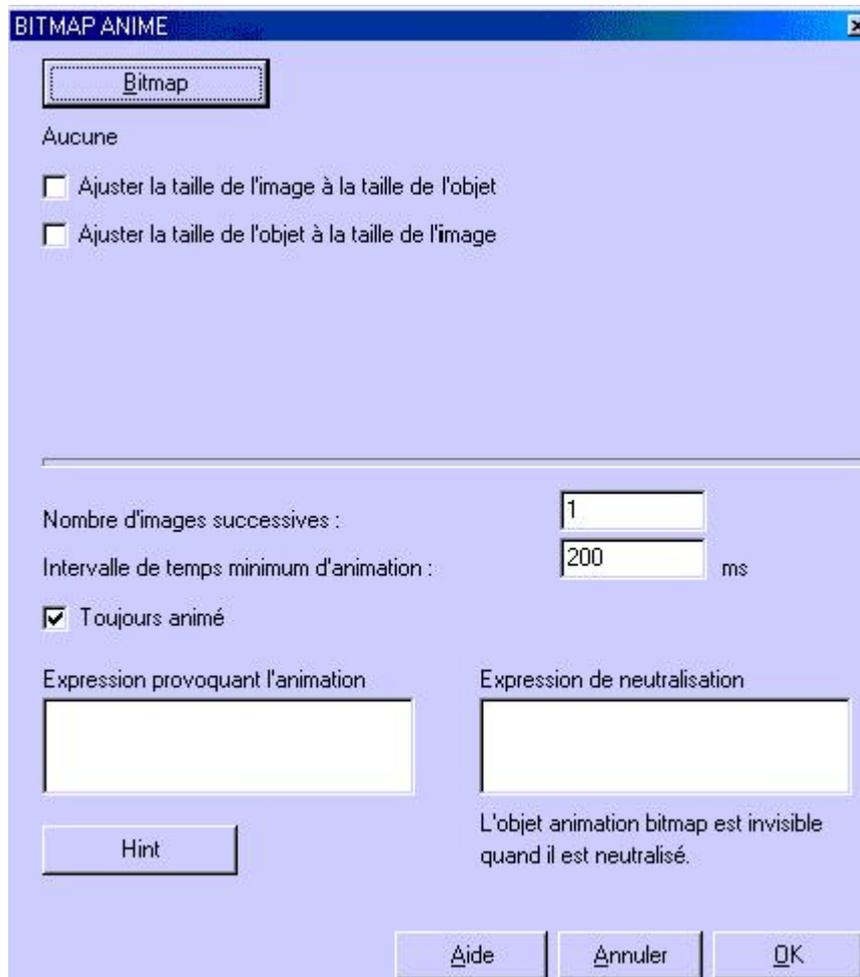
Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.7.4 Animated bitmap

This Object permits to create autonomous animations on supervisory forms. Has the object animation contrarily, there is not notion of state in this object.

It is necessary for this object to create a bitmap that will contain all pictures of the animation glued the some to the other in a horizontal way. It is possible to consult the project of example " Animate bitmap ". This project is accessible in the directory " \Program Files\Optimalog\Apigraf IP\Exemples ".

Dialog box Object animate Bitmap



The parameters are the next one :

" Button Bitmap "

This control permits to select the picture bitmap of the animation, this bitmap contains the different pictures of the animation glued the some to the other in a horizontal way. The project of example " Animate bitmap " illustrate the decomposition of the picture bitmap.

" To adjust the size of l 'picture to the size of the object "

When this option is activated, the size of the picture is always the same that the one of the object. The fact to enlarge the object will have therefore the effect of zooming in the picture.

" To adjust the size of the object to the size of the picture "

Permits to have the picture in real size with the size of the object adjusted to the one of the picture

" Numbers of successive pictures "

Specify the total number of pictures of the animation present in the picture bitmap. If the entered number is not correct, the animation won't be correct. It is this parameter that permits to cut the pictures of the animation in the picture bitmap.

" Option Always animate "

If this option is checked, the object will always be in animation at the time of the execution of the containing supervisory form the object.

" Expression provoking the animation "

Permits to specify a boolean expression that will trigger the animation when this one will be equal in TRUE. In this case, it is necessary to uncheck the option " Still animate ".

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

" Expression of neutralization "

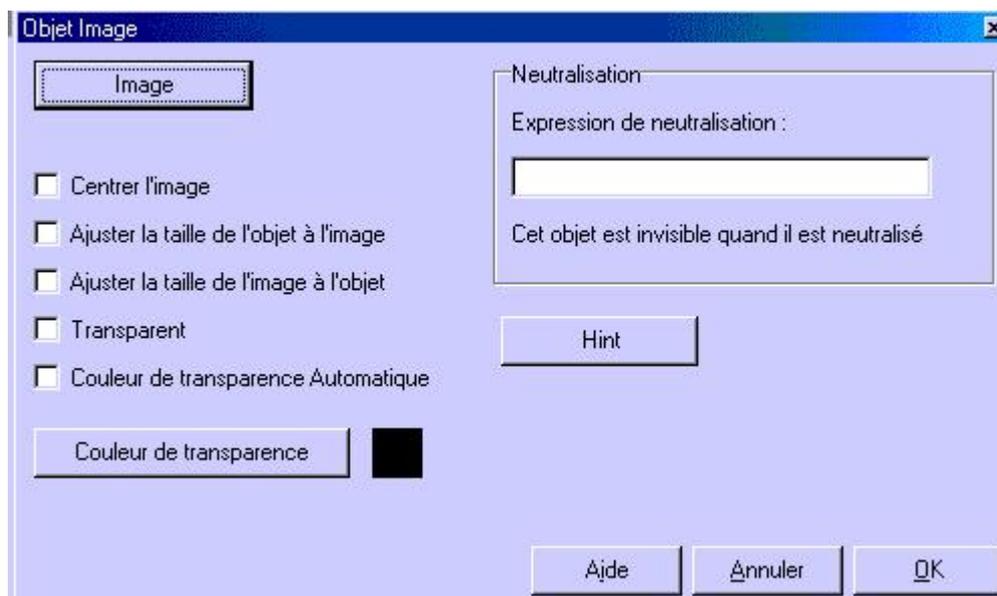
Permits to specify a boolean expression permitting to make invisible the object animation when this one will be equal in TRUE.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.7.5 Picture

This object permits to display a picture.

Dialog box parameter Picture



The following options are available :

Picture

Permits to specify the corresponding file to the picture. The following formats are admitted: * .BMP; * .EMF; * .WMF; * .ICO

Remark : A collection of pictures to the format BITMAP is available in the APIGRAFIP\GRAPHIQUES\BITMAPS

subdirectory

Option to Center the picture

This option permits to center the picture in the object.

Option to Adjust the size of the object to the picture

This option automatically adjusts the size of the object creates to the real size of the picture.

Option to Adjust the size of the picture to the object

This option automatically adjusts the size of the picture to the real size of the object. The fact of redimensionner the object will have an influence therefore on the size of the picture.

Transparent

Permits to define if one wants that the picture has a color of transparency. So such is the case, the color of the picture defined like color of transparency won't be displayed. The picture lets appear what is behind plane to the level of the transparency color then.

Automatic transparency color

If this option is checked, the color of transparency is automatically determined. It is in principle the color of bottom of the picture that is taken in account.

Color of transparency

This option permits in the case or the option " Automatic transparency color " is not checked to choose the color of transparency of the picture.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The picture will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.8 GAUGES tab objects

3.2.6.8.1 Gauge

The object sizes up permits to visualize the state of a numeric type variable graphically. It doesn't permit the modification of this variable however.

Dialog box parameter Object Sizes up

The options are the following :

Expression

To indicate in this field the variable PLC to use (of type numeric integer Example: R3)

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The tonnage will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Color Blocks

Permits to choose the color of the bar of the tonnage.

Color Melts

Permits to choose the color of bottom of the tonnage.

Type Shape

This option permits to choose the shape of the tonnage (horizontal or vertical bar, pie chart or dial).

Option Display percentage

This option permits to display the percentage of replenishment of the tonnage in relation to the values min and max.

Value min

Defines the value below which the tonnage is to its minimal position.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R20

In this case, the boundary-mark min of the tonnage will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value max

Defines the value above which the tonnage is to its maximal position.
This field offers the same options of use that the field " Value Min " here over.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.8.2 Vu meter

This object has some properties similar to an object sizes up.

Only its aspect of seen meter to leds differentiates it of this object. It permits to visualize a numeric variable (Example a register: R12) :

Aspect of the object seen meter**Dialog box object seen meter**

The parameters are :

" Expression "

To indicate in this field the name of the variable to control. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Value Min "

This option defines the value below which the cursor is to its minimal position.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

In this case, the boundary-mark min of the scroll bar will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Value Max "

This option defines the value above which the cursor is to its maximal position.

This field offers the same options of use that the field " Value Min " here over.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Orientation "

This option permits to choose the sense of display of the cursor, vertical or horizontal.

" Width of the leds "

Permits to define the width of each of the leds composing seen it meter.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). Seen it meter will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

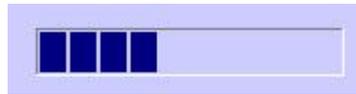
« **Hint** »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.8.3 Progress bar

This object permits to control a bar of progression with a variable PLC.

Seen of a progression rod :



Dialog box " Rod of progression "

The fields are the next one :

" **Expression** "

To indicate in this field the name of the variable PLC that must control the bar of progression. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" **Orientation** "

This option permits to choose the sense of display of the progression bar, vertical or horizontal.

" **Value Min** "

This option defines the value for which the bar of progression is to its minimum.

It is possible to specify in this field a constant value or the expression of an integer type variable PLC :

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

In this case, the boundary-mark min of the progression bar will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

" Value Max "

This option defines the value for which the bar of progression is maximal.

This field offers the same options of use that the field " Value Min " here over.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The bar of progression will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

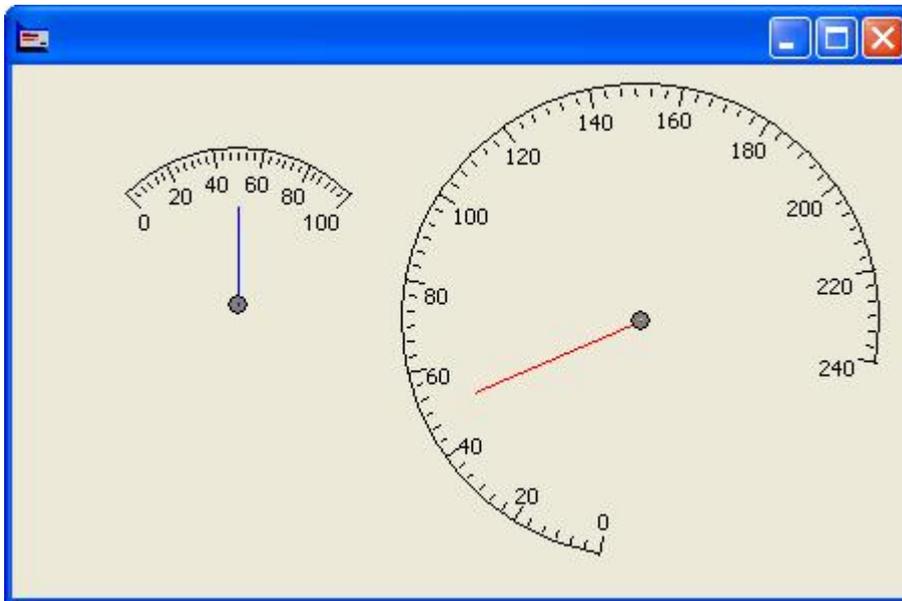
Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.8.4 Simple circular gauge

The object "Sizes up circular simple" permits to visualize the state of a numeric type variable graphically. It doesn't permit the modification of this variable.

Example of simple circular tonnage:



Dialog box " simple circular tonnage " - General tab:

Jauge circulaire simple

Général Neutralisation Graduations

Expression :

Valeur Min :

Valeur Max :

Couleur de l'aiguille

Hint

Angle de départ du cadran : degrés

Angle total du cadran : degrés

Aide Annuler OK

The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize in Seen it meter to needle. This variable must be of numeric type.

Example : R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Min

This parameter indicates the value of the lower boundary-mark of the graduating. This parameter must be of numeric type.

Example : 100

R13

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Max

This parameter indicates the value of the boundary-mark superior of the graduating. This parameter must be of numeric type.

Example : 1000
R14

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Color of the needle

Permits to select the color of the needle of the circular tonnage.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Angle of departure of the dial

Specify to what value of angle begins the dial. It permits to position the dial below for example.

Total angle of the dial

Specify the angular total value in degrees of the dial. To have a dial on a semi-circle, to enter the value 180

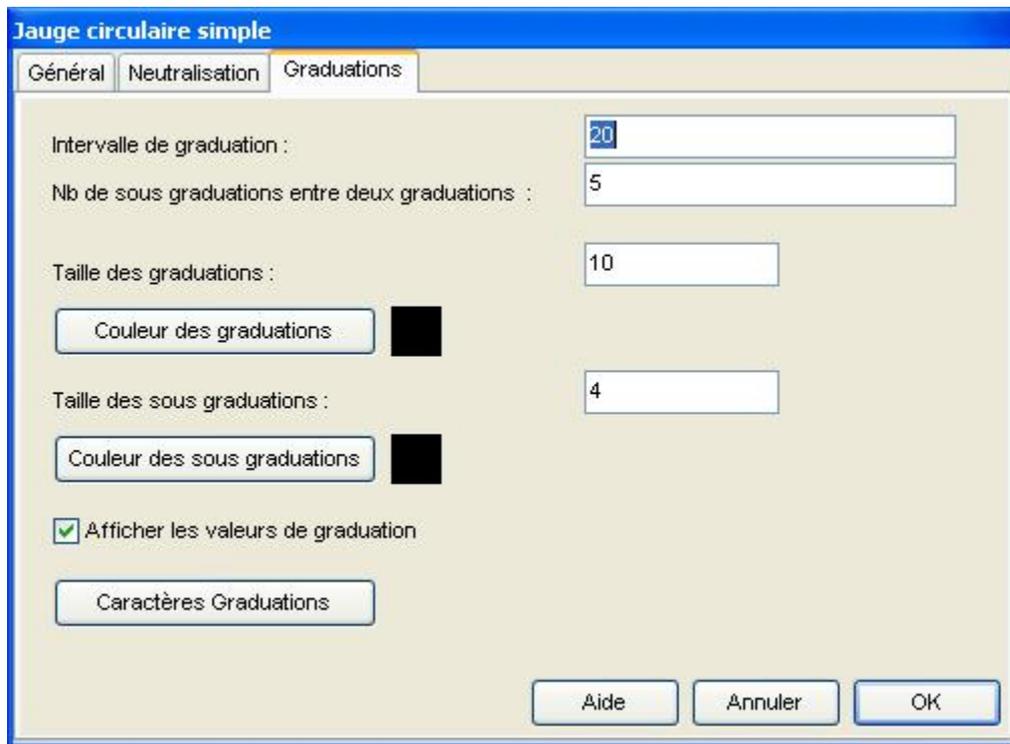
Dialog box " simple circular tonnage" - Tab Neutralization :**Expression of neutralization**

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Dialog box " simple circular tonnage" - Tab Graduating :



This tab permits to define the display of the graduating on the circular tonnage :

Interval of graduating :

Defines the interval between 2 graduating. This value must be of numeric type.

Nb of coins graduating between 2 graduating

Defines the number of reference marks displayed between 2 graduating

Size of the graduating

Defines the length of the graduating in pixels

Button "Color of the graduating"

Defines the color of the dashes that symbolizes the graduating.

Size of the under-graduating

Defines the length of the under-graduating in pixels

Button "Color of the under-graduating"

Defines the color of the dashes that symbolizes the under-graduating.

Option to "Display the values of graduating"

If this option is checked, the values numeric correspondantes to the graduating will be displayed.

Button "Characters graduating"

This button permits to define the police of display of the values of graduating.

3.2.6.8.5 Advanced Circular gauge

The object "Sizes up circular" permits to visualize the state of a numeric type variable graphically. It doesn't permit the modification of this variable.

Circular tonnage example :



Dialog box " circular tonnage " - General tab:

The dialog box 'JAUGE CIRCULAIRE' has the following elements:

- Tabbed interface with 'Général' selected.
- Input fields:
 - Expression:
 - Valeur Min:
 - Valeur Max:
- A 'Hint' button.
- Bottom buttons: 'Aide', 'Annuler', and 'OK'.

The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize in Seen it meter to needle. This variable must be of numeric type.

Example : R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Min

This parameter indicates the value of the lower boundary-mark of the graduating. This parameter must be of numeric type.

Example : 100
R13

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Max

This parameter indicates the value of the boundary-mark superior of the graduating. This parameter must be of numeric type.

Example : 1000
R14

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

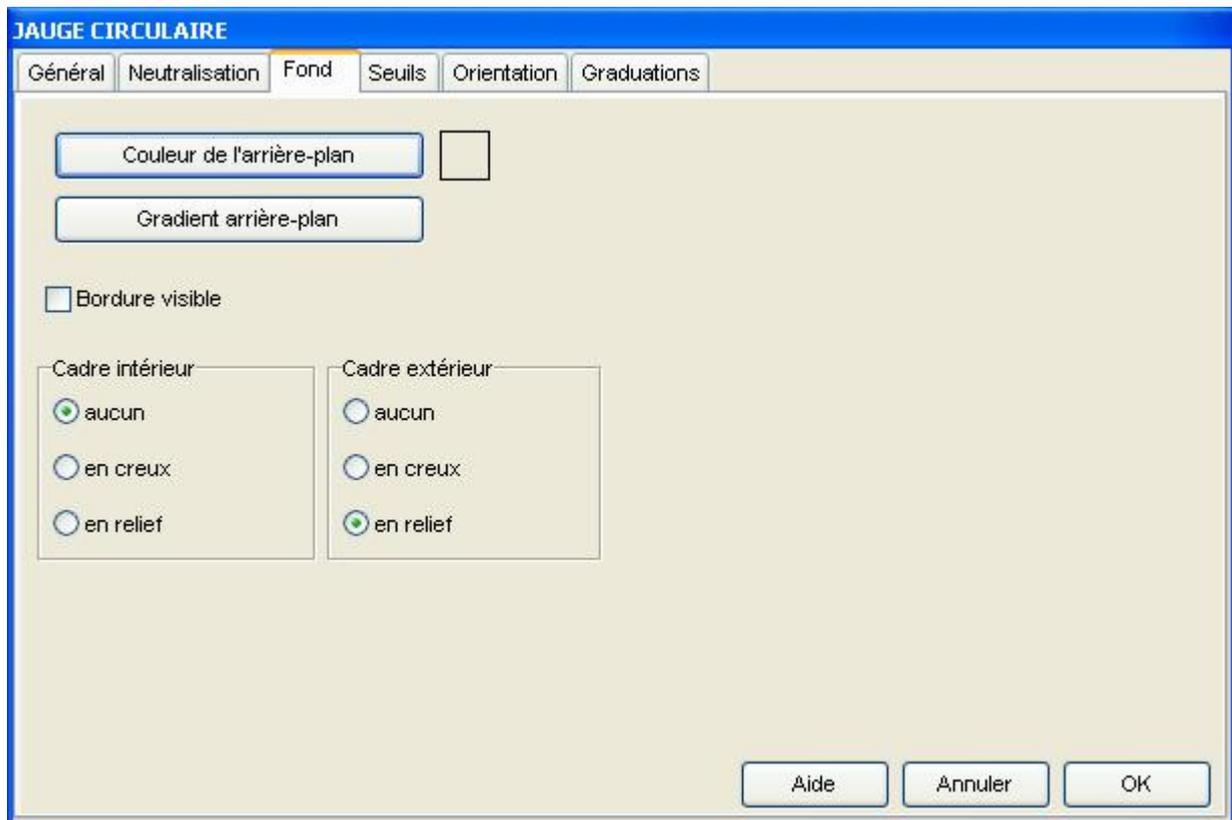
Dialog box " circular tonnage " - Tab Neutralization :**Expression of neutralization**

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

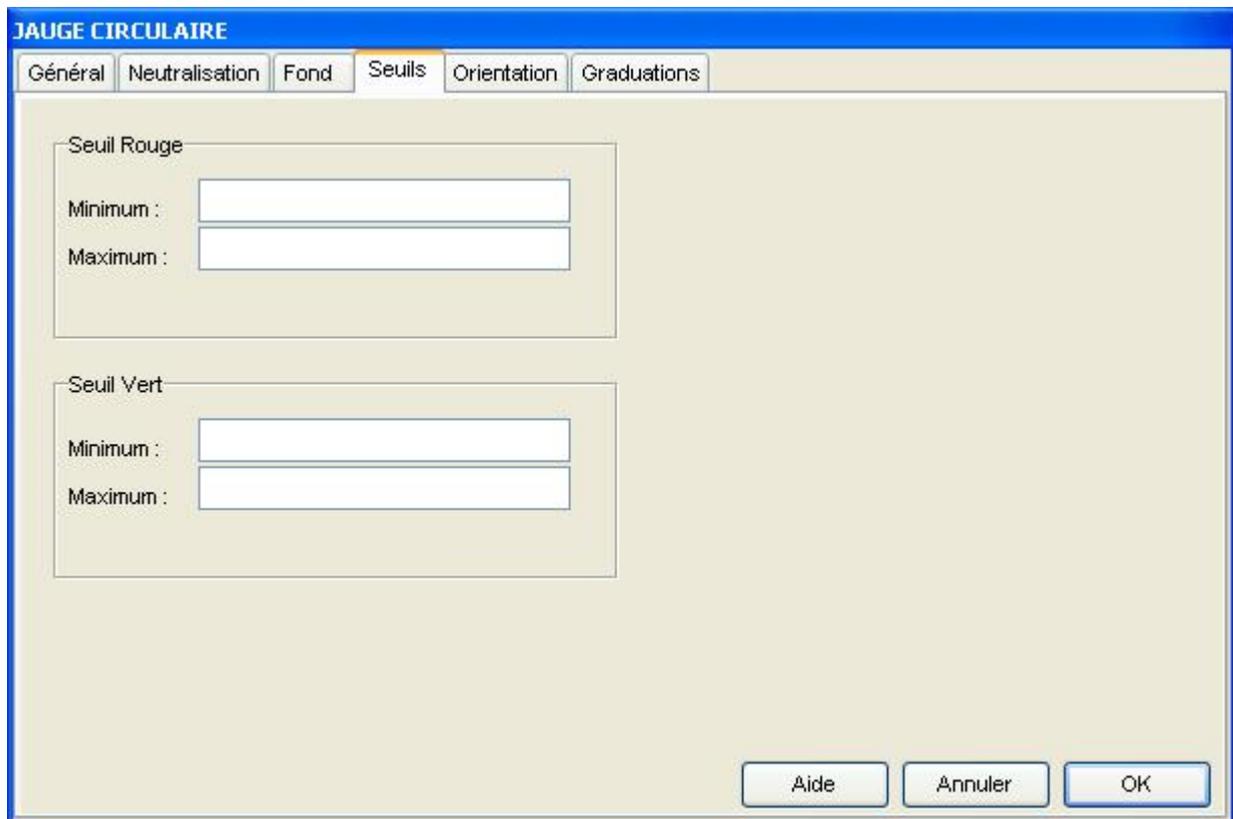
Dialog box " circular tonnage " - Tab Melts :



This tab permits to define the color and the aspect of the setting that contain the circular tonnage.

Dialog box " circular tonnage " - Tab Doorsteps :

This tab permits to parameterize two zones marked on the tonnage, a green zone and a red zone.



For every red and green doorstep, the values min and max are paramétrables. To enter values of constant or variable numeric types.

Examples : R10

100
0

Dialog box " circular tonnage " - Tab Orientation :

Angle of departure :

Permits to define the angle of departure of the dial of the tonnage

Total angle of the tonnage :

Permits to define the angular beach of display of the tonnage.

Dialog box " circular tonnage " - Tab Graduating :

This tab permits to define the display of the graduating on the circular tonnage :

Interval of graduating :

Defines the interval between 2 graduating. This value must be of numeric type.

Nb of coins graduating between 2 graduating

Defines the number of reference marks displayed between 2 graduating

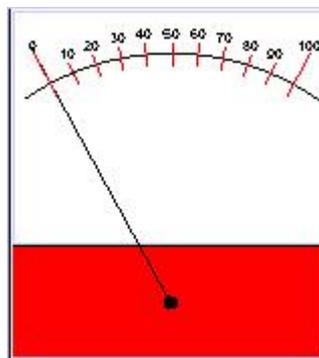
Distance of the coins graduating

Defines the distance between the coins graduating and the dial. The value of integer numeric type is given in number of pixels.

3.2.6.8.6 Analog Vu meter

The object Seen meter to needle permits to visualize the state of a numeric type variable graphically. It doesn't permit the modification of this variable however.

Example of Seen meter to needle :



Dialog box " Seen meter to needle " :

The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize in Seen it meter to needle. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text"

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Color Melts

Specify the color of bottom of Seen it meter to needle.

Color Bases

Specify the color of the low part of Seen it meter to needle.

Color Directs

Specify the color of the needle.

Scale Min

This parameter indicates the value of the lower boundary-mark of the graduating of Seen it meter to needle. This constant parameter can be of integer type or decimal

Example : 10
0,5

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Scale Max

This parameter indicates the value of the boundary-mark superior of the graduating of Seen it meter to needle. This constant parameter can be of integer type or decimal

Example : 100
10,05

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Interval graduating

This parameter indicates the value separating two graduating. This constant parameter can be integer or decimal.

Example: 10
1,5

Number of decimals

The graduating can be displayed with decimals. This parameter indicates the number of decimals displayed. This parameter is of integer type.

Button Character Graduating

Allows the access the dialog box concerning the type of display of the graduating (color, police.).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.9 CURSORS / TRACKBARS tab objects

3.2.6.9.1 Trackbar

This object permits to control a numeric variable with the help of a cursor, whose displacement provokes a proportional variation of the variable.

Dialog box parameter Cursor

The input fields of this box are the next one :

Expression

To indicate in this field the name of the variable to control. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Orientation

This option permits to define if the cursor must be oriented horizontally or straight up.

Progression

Permits to specify the sense of progression of the cursor, that is to say the sense in which the value associated to the cursor grows.

If the cursor is oriented horizontally, one can choose a progression from right to left or from left to right. If the cursor is oriented vertically, one can choose a progression from top to bottom or of bottom in top.

Value Min

This option defines the value below which the cursor is to its minimal position.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : R20

For the variable PLCs, one can specify the address besides distant IP of the kernel APIGRAF PLC in the expression. If it is the case, it is this address that will be taken in account for the cooling of the value Min.

Example : !193.0.1.1 !R12

In this case, the boundary-mark min of the cursor will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Max

This option defines the value above which the cursor is to its maximal position.

This field offers the same options of use that the field " Value Min " here over.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Graduating

Specify the resolution of the graduating that is bound to the cursor to the level of the display.

Zone selected : Beginning and End

This option permits to display a certain zone of the cursor in blue.

Position graduating

Permits to specify the place where is displayed the graduating associated to the cursor. If orientation is vertical, one can choose a position on the right or on the left. If orientation is horizontal, one can choose a position in top or below.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The cursor will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is checked, the cursor will be visible when it will have neutralized. However, the operator won't be able to act over.

If this option is not checked, the cursor will be invisible when it will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

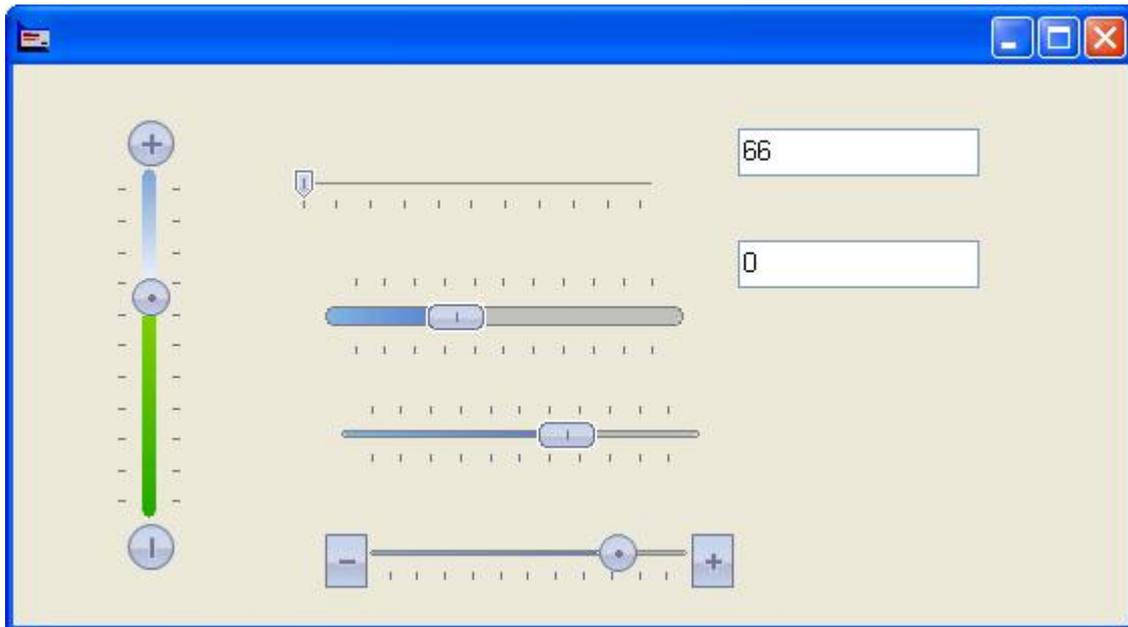
« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.9.2 Advanced TrackBar

This object permits to control a numeric variable with the help of a cursor, whose displacement provokes a proportional variation of the variable.

The drawing of this cursor is evolved more than the one of the classic cursor and is besides paramétrable.



Dialog box parameter Cursor - GENERAL Tab

CURSEUR / TRACKBAR

Général Neutralisation Boutons Curseur Glissière Fond

Expression :

Valeur Min :

Valeur Max :

Intervalle graduations :

Position graduations :

Progression :

Orientation

Horizontale

Verticale

Aide Annuler OK

The input fields of this box are the next one :

Expression

To indicate in this field the name of the variable to control. This variable must be of numeric type.

Example : R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Orientation

This option permits to define if the cursor must be oriented horizontally or straight up.

Progression

Permits to specify the sense of progression of the cursor, that is to say the sense in which the value associated to the cursor grows.

If the cursor is oriented horizontally, one can choose a progression from right to left or from left to right. If the cursor is oriented vertically, one can choose a progression from top to bottom or of bottom in top.

Value Min

This option defines the value below which the cursor is to its minimal position.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : R20

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Max

This option defines the value above which the cursor is to its maximal position.

This field offers the same options of use that the field " Value Min " here over.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Interval Graduating

Specify the resolution of the graduating that is bound to the cursor to the level of the display.

Position graduating

Permits to specify the place where is displayed the graduating associated to the cursor. If orientation is vertical, one can choose a position on the right or on the left. If orientation is horizontal, one can choose a position in top or below.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« **Hint** »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Dialog box parameter Cursor - Tab NEUTRALIZATION

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The cursor will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is checked, the cursor will be visible when it will have neutralized. However, the operator won't be able to act over.

If this option is not checked, the cursor will be invisible when it will have neutralized.

Dialog box parameter Cursor - Tab BUTTONS

This cursor offers the possibility to position some buttons to every extremity of the cursor. These buttons will act on the cursor in the sense more and in the sense less.

Option " Let's put extremity visible "

If this option is checked, the buttons will be visible to every extremity.

Shape of the buttons :

To choose here the shape of the buttons (Round or Square)

Dialog box parameter Cursor - Tab CURSOR

Shape of the cursor

This field permits to select the shape of the cursor: Drawer, Rectangle, Round, rounded Rectangle,

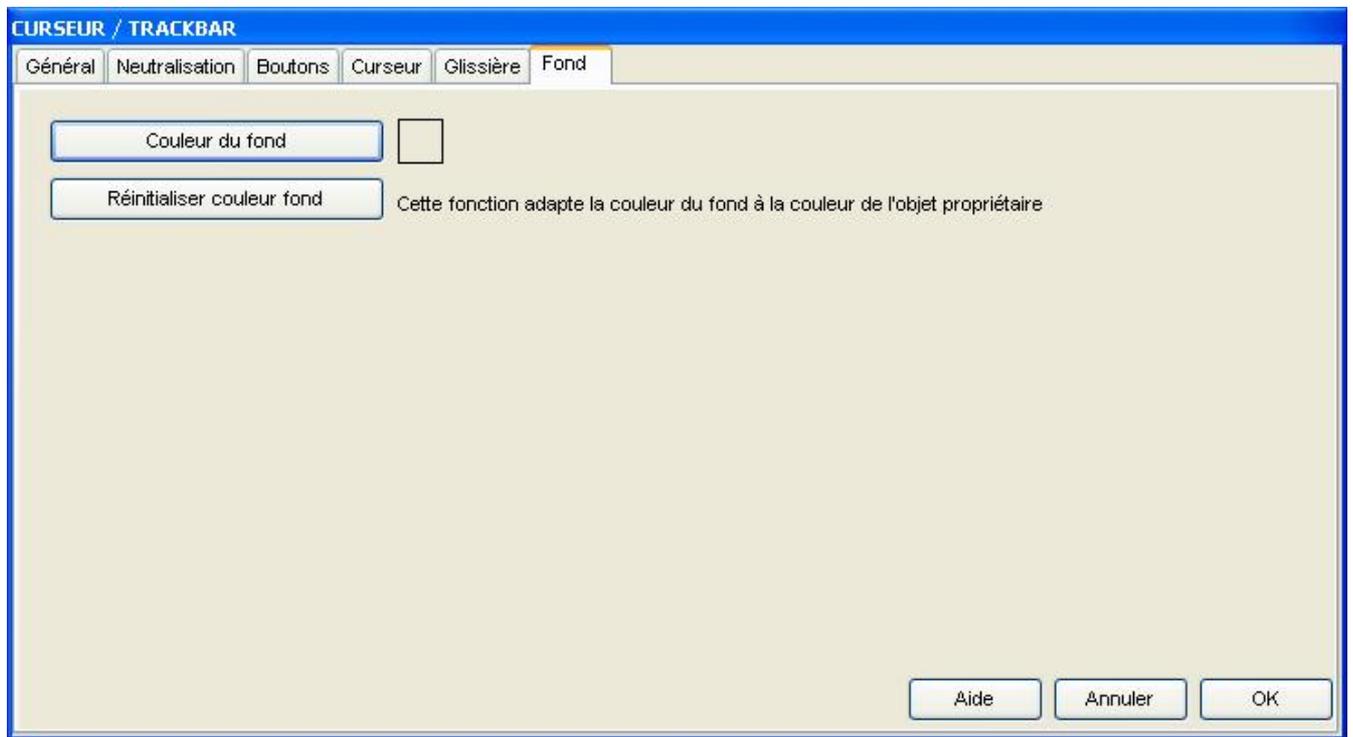
Size and width of the cursor

These 2 fields permit to proportion the cursor.

Dialog box parameter Cursor - Tab GROOVE

This tab permits to parameterize the aspect of the groove of the cursor.

Dialog box parameter Cursor - Tab MELTS



Button "Color of bottom"

Permits to choose the color of bottom of the cursor.

Button to "Reset the color of bottom"

Permits to initialize the color of bottom with the color of bottom of the object owner.

3.2.6.9.3 Scroll bar

The object scroll bar is a variant of the object cursor. Its aspect is the one of the scroll bars standard of Windows®. This object permits to make vary a variable PLC of integer numeric type (register).

Dialog box parameter scroll bar

The fields are the next one :

Expression

To indicate in this field the name of the variable to control. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Orientation

This option permits to choose the sense of display of the cursor, vertical or horizontal.

Progression

Permits to indicate in what sense the value of the variable increases.

Value min

This option defines the value below which the cursor is to its minimal position.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

In this case, the boundary-mark min of the scroll bar will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value max

This option defines the value above which the cursor is to its maximal position.

This field offers the same options of use that the field " Value Min " here over.

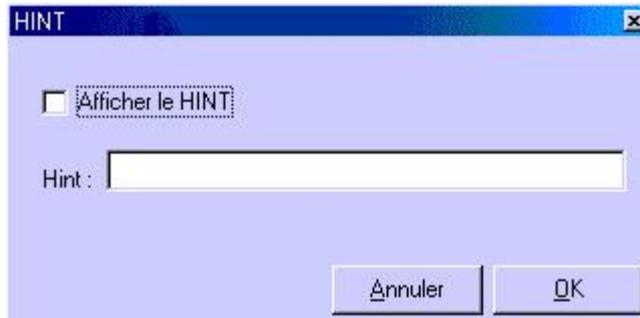
It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The scroll bar will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is checked, the scroll bar will be visible when it will have neutralized. However, the operator won't be able to act on the bar and this one will be grey.

If this option is not checked, the scroll bar will be invisible when it will have neutralized.

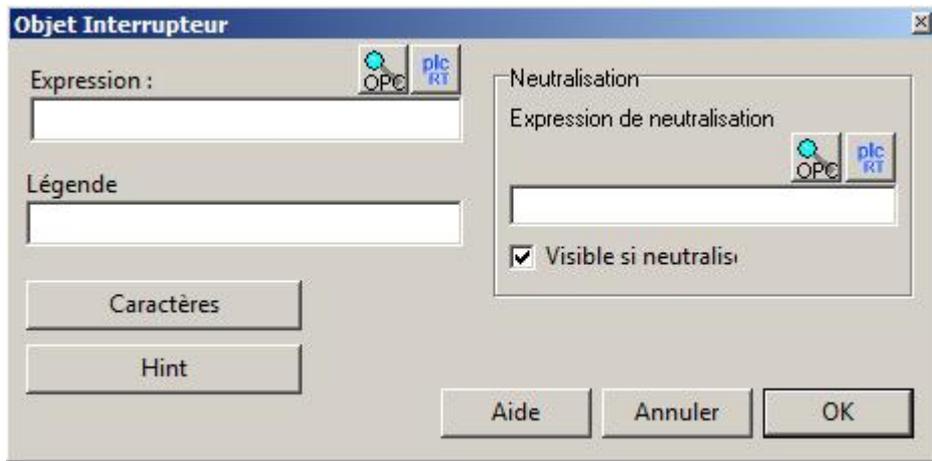
3.2.6.10 SWITCHES tab objects**3.2.6.10.1 ON/OFF Switch**

The object Button Works / Stop simulates a switch. The appearance of this object varies with a click on the left button of the mouse above the object or according to the value of its control variable.

This variable can be of boolean type (L flag, logical Switch SA) or integer (Register R).

Picture of the button :



Dialog Box « ON/OFF Switch s » :

Les paramètres sont les suivants :

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The button will have neutralized every time then that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " Visible if neutralized "

If this option is active, the button will be visible when it will have neutralized but this one won't be able to be manipulated. Besides the text of the button will appear in grey tints.

If this option is not active, the button will be invisible as soon as the expression of neutralization is to TRUTH (So of course, the expression of neutralization is defined).

Legend

Insert a text legend a the top of the object

Font Button

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Option « visible si neutralisé »

Si cette option n'est pas cochée, l'interrupteur deviendra invisible lorsqu'il sera neutralisé.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



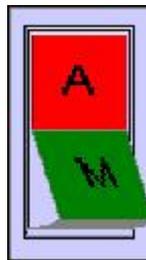
The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text.



It dialog box type is common to all switches, please refer you to " Dialog box lever with color "

3.2.6.10.2 Lever switch #2

The object Lever without color simulates a switch. The appearance of this object varies with a click on the left button of the mouse above the object or according to the value of its control variable. This variable can be of boolean type (L flag, logical Switch SA) or integer (Register R).

Picture of the lever :



It dialog box type is common to all switches, please refer you to " Dialog box [ON/OFF SWITCH](#) ²²³"

3.2.6.10.3 Emergency stop button

The object Button marks stop of emergency simulates a switch.

The appearance of this object varies with a click on the left button of the mouse above the object or according to

the value of its control variable. This variable can be of boolean type (L flag, logical Switch SA) or integer (Register R).

Picture of the button marks stop of emergency :



Its dialog box type is common to all switches, please refer you to " Dialog box [ON/OFF SWITCH](#) ^[223]"

3.2.6.11 SELECTORS tab objects

3.2.6.11.1 2/3 positions Selector

L'objet Sélecteur 2 / 3 positions simule un sélecteur pouvant avoir 2 ou 3 états.

L'apparence de cet objet varie avec un clic sur le bouton gauche de la souris au-dessus de l'objet ou en fonction de la valeur de sa variable de contrôle. Il permet de faire évoluer la variable de 0 à 2 avec un cycle de positions : 0 – 1 – 2 – 1 – 0.

Exemple de Sélecteur 3 positions avec étiquette:



Exemple de Sélecteur 3 positions sans étiquette:



Boîte de dialogue « Sélecteur 2 / 3 positions »

Les paramètres sont les suivants :

Expression

Permet de définir le nom de la variable de contrôle de l'objet (variable associée à l'objet Sélecteur 3 positions). Cette variable sert d'index pour chaque état de l'animation (un état correspond à une valeur de la variable). Indiquer dans ce champ la variable automate à utiliser (de type numérique entier)

Exemple : R20

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !R20

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Etiquette

Permet de mettre une légende au-dessus de l'objet sur une ou deux ligne. Il est possible de spécifier une police de caractères pour chaque ligne/

Option 2 positions

Si cette options est coché le sélecteur aura 2 positions.

Option 3 positions

Si cette options est coché le sélecteur aura 3 positions.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The button will have neutralized every time then that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory

form with parameters ")

Option " Visible if neutralized "

If this option is active, the button will be visible when it will have neutralized but this one won't be able to be manipulated. Besides the text of the button will appear in grey tints.

If this option is not active, the button will be invisible as soon as the expression of neutralization is to TRUTH (So of course, the expression of neutralization is defined).

Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

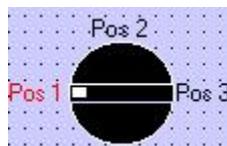
Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

The object 2/3 positions selector simulate a selector capable to have 2 or 3 state.

The appearance of this object varies with a click on the left button of the mouse above the object or according to the value of its control variable. It permits to make evolve the variable of 0 to 2 with a cycle of positions: 0 . 1 . 2 . 1 . 0.

Example of Selector 3 positions :



Dialog box " Selector 2 / 3 positions "

The parameters are the next one :

Expression

Permits to define the name of the variable of control of the object (variable associated to the object Selector 3 positions). This variable acts as index for every state of the animation (a state corresponds to a value of the variable). To indicate in this field the variable PLC to use (of integer numeric type)

Example : R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R20

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text"

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Option 2 positions

If this options are checked the selector will have 2 positions.

Option 3 positions

If this options are checked the selector will have 3 positions.

Button " Color Selector "

Specify the color of the selector.

Position 1

Permits to put a text that will be attached to the position 1 and will become of white color when the selector will be at the state 1.

Position 2

Permits to put a text that will be attached to the position 2 and will become of white color when the selector will be at the state 2.

Position 3

This parameter is visible that if the " 3 positions " is checked.

Permits to put a text that will be attached to the position 3 and will become of white color when the selector will be at the state 3.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the selector will become invisible when it will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.11.2 2/3 positions Selector Without Label

L'objet Sélecteur sans étiquette 2 / 3 positions simule un sélecteur pouvant avoir 2 ou 3 états. C'est le même objet que le Sélecteur 2 / 3 positions mais il ne possède pas d'étiquette.

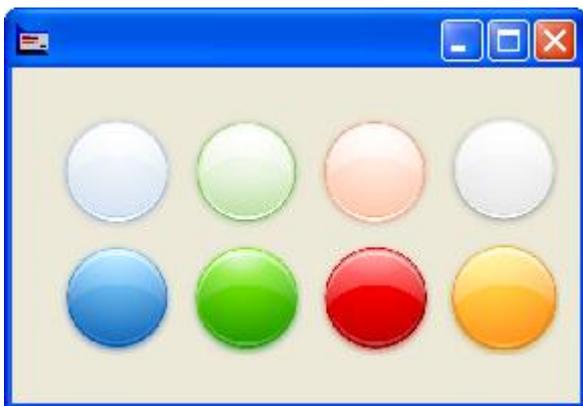
See: [2/3 positions Selector](#)^[226]

3.2.6.12 INDICATORS tab objects

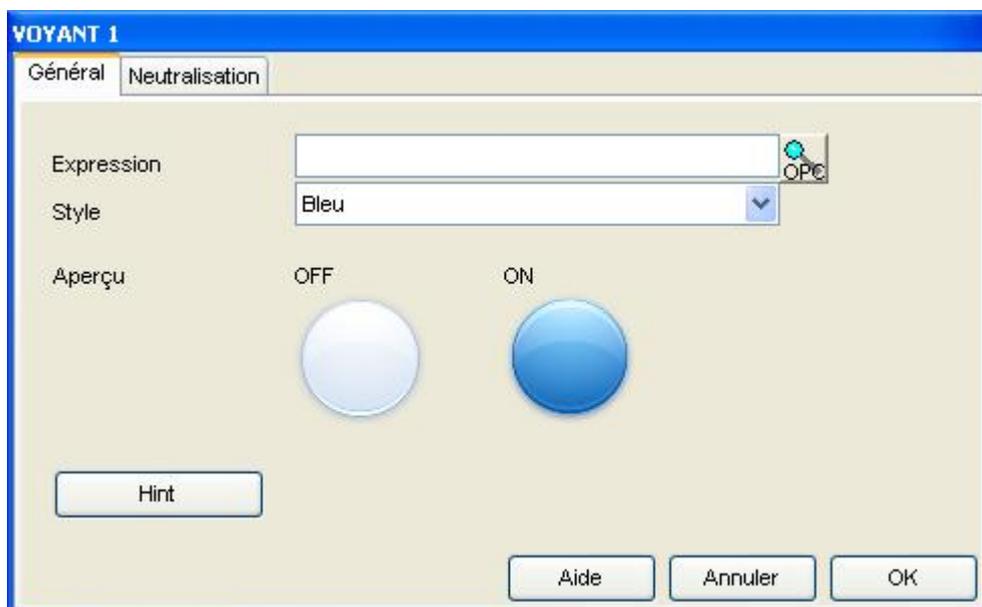
3.2.6.12.1 Indicator type 1

This object simulates a seer 2 states: white to rest, quick color in working (color to define: blue, green, red or yellow).
Its state varies according to its variable of control (0 / 1).

Example of Seer type 1 :



Limp of dialog " Seeing type 1 " - General tab :



The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize with the seer. This variable must be of boolean type (that is to say to two states 0 and 1).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

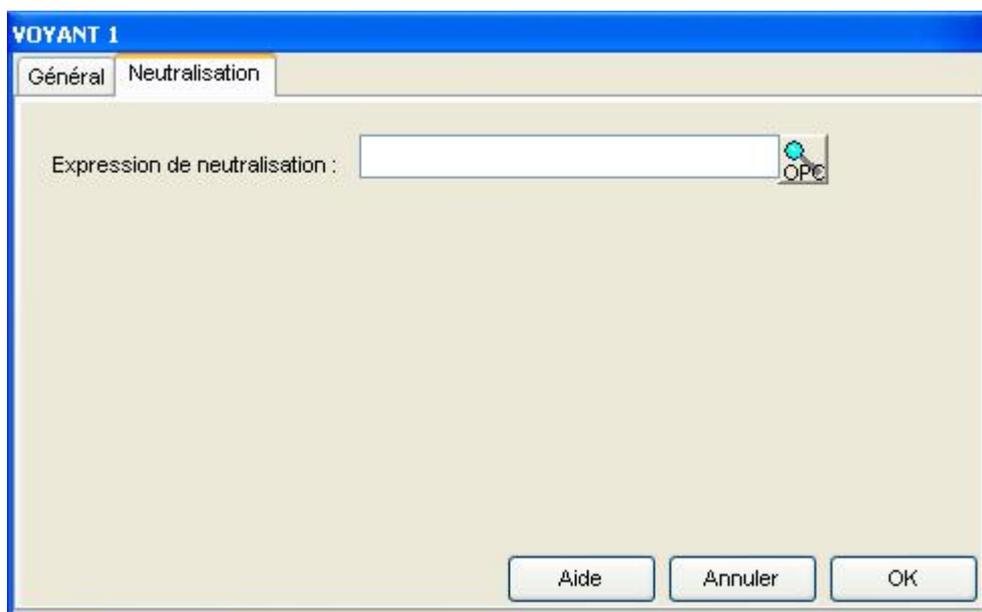
It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If the variable comes from an OPC server, the OPC button permits to explore the OPC server and to find the variable to supervise in an explorer.

Style

Permits to define the seer's appearance: Blue, green, red or yellow.

Limp of dialog " Seeing type 1 " - Tab Neutralization:



Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

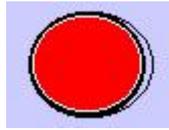
It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.12.2 Simple indicator

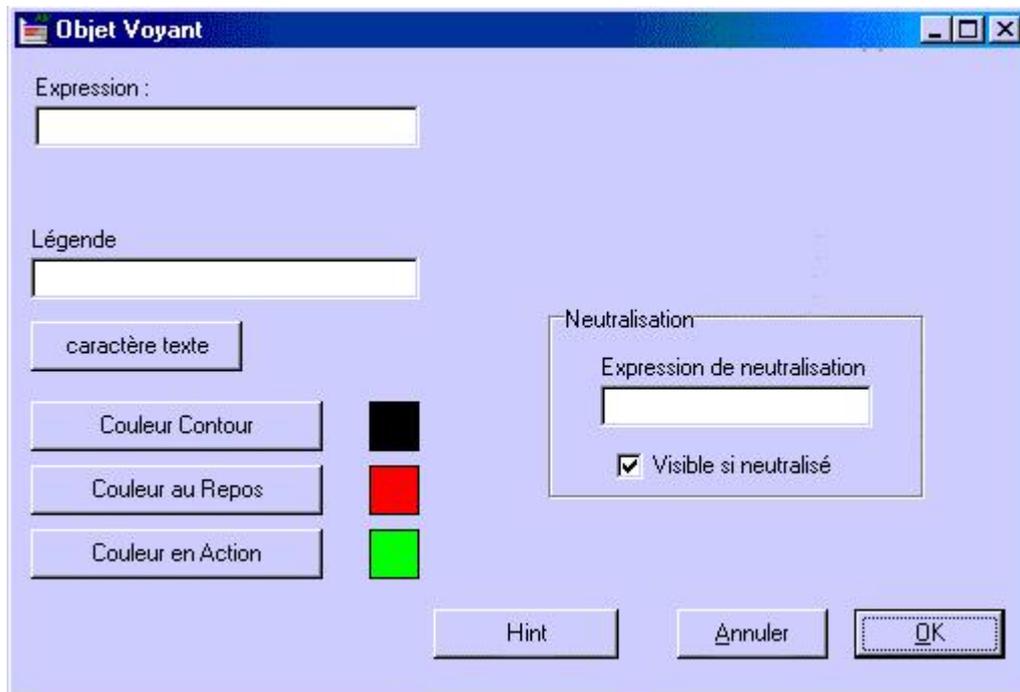
This object simulates a seer 2 states: red to rest, green in working (color by default).

Its state varies according to its variable of control (having 0 / 1 or False / True as value).

Simple Seer example :



Lim of dialog " Seeing simple " :



The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize with the seer. This variable must be of boolean type (that is to say to two states 0 and 1).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If the variable comes from an OPC server, the OPC button permits to explore the OPC server and to find the variable to supervise in an explorer.

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text"

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Color Contour

Specify the color of the seer's contour.

Color to Rest

Specify the seer's color to rest (variable = 0 or false).

Color in Action

Specify the seer's color in action (variable = 1 or True).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the seer will become invisible when he will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

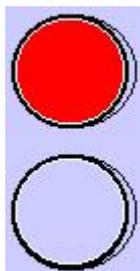
Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.12.3 Double indicator

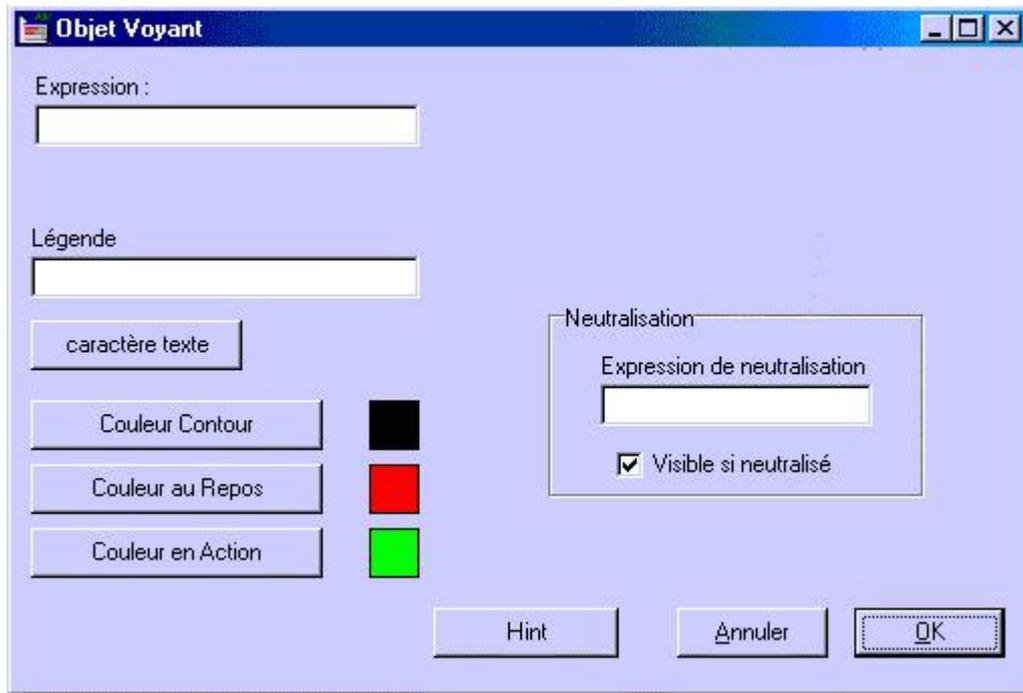
This object simulates two seer with only one variable: the first is of red color to rest and gray in action, the second is of gray color to rest and green in action (Color by default).

Its state varies according to its variable of control (having 0 / 1 or False / True as value).

Double Seer example :



Limp of dialog " Seeing double " :



The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize with the seer. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text"

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Color Contour

Specify the color of the seer's contour.

Color to Rest

Specify the color of the first seer to rest (variable = 0 or false).

Color in Action

Specify the color of the second seer in action (variable = 1 or True).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

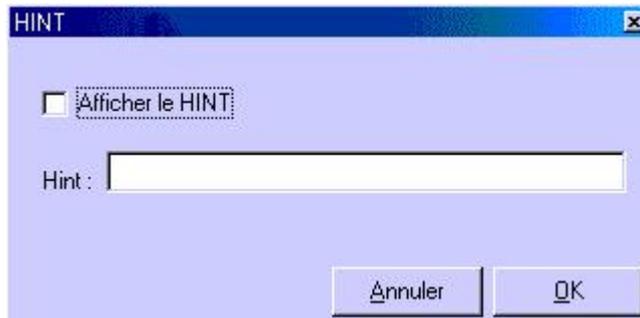
If this option is not checked, the seer will become invisible when he will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

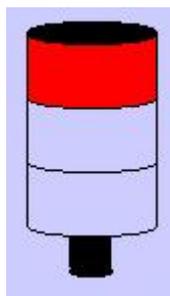
Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.12.4 3 colors column indicator

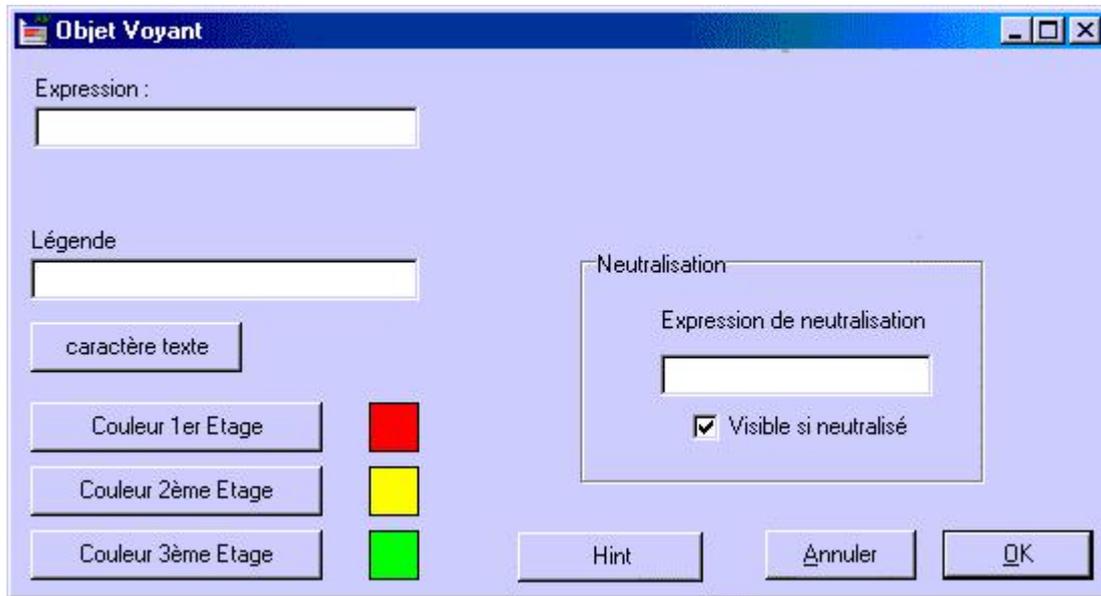
This object simulates a column constituted of three controlled levels with only one variable: the first floor is active for a value hopeless of the variable, its color is by default the red, the second is active for a value equal to 1, its color is by default the yellow and the third floor is active for a value equal to 2, its color is by default the green.

The floors non active take are grey.

Example of Column 3 colors :



Limp of dialog of " Seeing double " :



The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize with the column. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text"

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Color 1st floor

Specify the color of the high part of the column, activate when the variable is hopeless

Color 2nd floor

Specify the color of the central part of the column, activate when the variable is equal to 1.

Color 3rd floor

Specify the color of the low part of the column, activate when the variable is equal to 2.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the column will become invisible when it will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

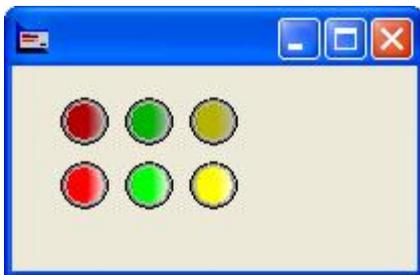
« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

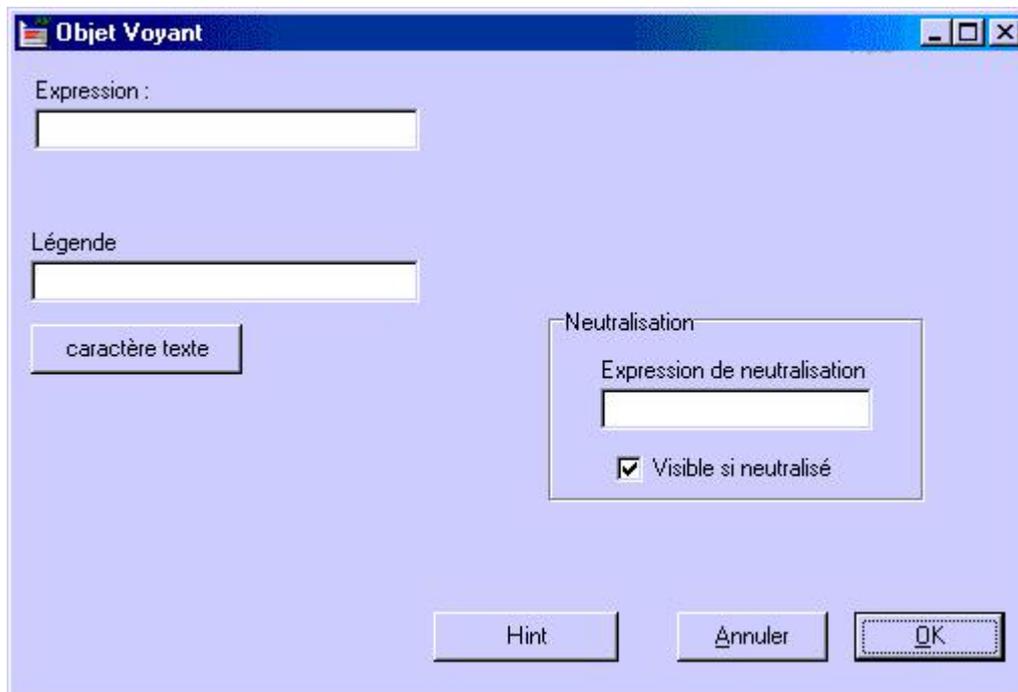
3.2.6.12.5 DEL indicator

This object is equivalent to a seer but with the appearance of a diode extinct électroluminescente or lit.

Examples of Leds :



Limp of dialog of " Led " :



The parameters are the next one:

Expression

To indicate in this field the name of the variable to visualize with the seer. This variable must be of boolean type (that is to say to two states 0 and 1).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

If the variable comes from an OPC server, the OPC button permits to explore the OPC server and to find the variable to supervise in an explorer.

Legend

Permits to put a legend above the object. This parameter only accepts the text.

Button " Characters text"

Allows the access the dialog box concerning the type of display of the legend (color, police.).

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the Led will become invisible when he will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the

object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.13 EXCEL tab objects

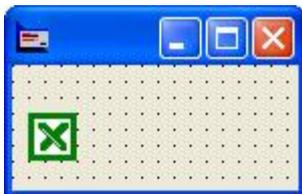
3.2.6.13.1 EXCEL link

The object EXCEL link permits to read and to write some variable PLCs since / toward an EXCEL folder. The update is triggered by a variable boolean PLC. When the value of this variable passes to 1, the EXCEL file is open, the spreadsheet is selected and the data are put up to date in reading or in writing. The value of the starting point variable is handed to 0 then.

The principle is the next one: One associates the variable PLCs coming from Optima PLC / APIGRAF or an OPC link to a cell of an EXCEL form and one specifies for every tie if this one is in writing or in reading. If the tie is then in reading the value will be read in the EXCEL cell and its value will be affected to the variable PLC correspondent. If the tie is then in writing the value will be read in the PLC and this value will be affected to the cell corresponding EXCEL.

All links in writing or in reading are executed only one time as soon as the value of the starting point variable passes to 1.

The object EXCEL link is only visible at conception on the supervisory form :



Dialog box " EXCEL link " :

Liaison Excel

Nom ou expression du fichier EXCEL : 'C:\TEST.XLS'
 (Si la valeur entrée n'est pas une variable, entrer le nom du fichier entre côtes '' : Exemple : 'C:\TEST.XLS')

Nom ou expression du classeur EXCEL : 'Feuil1'
 (Si la valeur entrée n'est pas une variable, entrer le nom du fichier entre côtes '' : Exemple : 'Feuil1')

Expression déclenchement liaison EXCEL : WSA100

Application EXCEL visible :
 Fermer EXCEL à la fin du transfert

Liste des liens en lecture/écriture

Aide Annuler OK

The parameters are :

- **Name or expression of the EXCEL file**

Specify the name of the EXCEL file to use. If the name of the file is fixed, to enter the name of the file between ratings :

Example : 'C:\REPEXCEL\TEST.XLS'

If the name of the file is variable, to enter the name of the variable (of string / chain type) whose value represents the name of the EXCEL file to use

Example : C10

- **Name or expression of the EXCEL folder**

Specify the name of the EXCEL folder to use in the file current EXCEL. If the name of the folder is fixed, to enter the name of the folder between ratings :

Example : 'Feuil1'

If the name of the folder is variable, to enter the name of the variable (of string / chain type) whose value represents the name of the EXCEL folder to use

Example : C10

The value by default of this field is: 'Feuil1'

- **Expression starting point EXCEL link**

Specify the variable PLC that is going to trigger the EXCEL liaison and therefore the update of the data in reading or writing.

The variable must be boolean and the starting point will be done at the time of the passage to 1 the value of the variable. The value of the variable will be handed to 0 at the end of the updates.

- **Application visible EXCEL**

Specify if the EXCEL application must be visible to the screen at the time of the link.

- **To close EXCEL at the end of the transfer**

Specify if the EXCEL application must be closed the done update once.

- **Cell Line Index Variable**

This field is optional. It specifies a numerical variable used to define the line index for some cells defined in the links without any line number.

If a cell is defined with a line number in the read/write links list (Example : A11) then the Line index variable will not be used. If the cell is defined without line number, then the Cell name will be completed with the current Line index variable value. If the cell name is "B" and the variable value is 2 then the name of the cell will be "B2".

It's possible to mix complete and not complete cell names in the link list.

- **Button "Read/write link list"**

Permits to define the list of the links. The button opens the following window :

Variable Automate	Cellule EXCEL	Lecture/Ecriture
WR100	A4	Ecriture
WR100+10	A5	Ecriture
WR100+20	B2	Ecriture

The window displays the list of the ties.

To add a tie, to click on the button "to Add", the following paramétrage window appears :

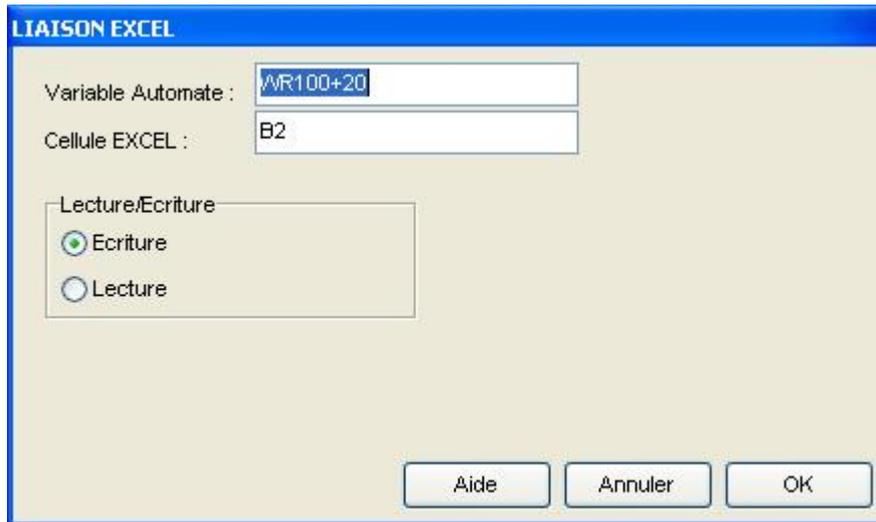
To inform the fields follows: Variable PLC (Exemple: R100), Cell EXCEL (Exemple: A4) and to choose the type of link (Writing or Reading).

To validate then with "OK"

To suppress a tie, to select the tie has suppress in the list with the help of the mouse, this one must appear then in subbrillance in the list and must click then on the button to "Suppress"

To modify a tie, to select the tie has modify in the list with the help of the mouse, this one must appear then in

surbrillance in the list and must click then on the button "to Modify." The window of parameterization opens up then :



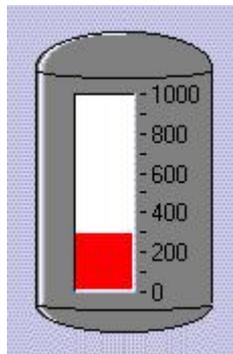
To modify the fields and to validate with "OK."

3.2.6.14 CUVE / FLOODGATE tab objects

3.2.6.14.1 Cuve

The object ferments permits to display a vat with an associated tonnage that the level of the vat symbolizes, it is of more possible to display on the vat a stepped up scale.

Example of vat :



Dialog box " Vat " :

The parameters are :

Expression

To indicate in this field the name of the variable to visualize in the vat. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Min

This option defines the value for which the vat is empty.

It is possible to specify in this field a constant value or the expression of a variable PLC :

Example : 1000
R20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

In this case, the boundary-mark min of the vat will be variable.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Value Max

This option defines the value for which the vat is full.

This field offers the same options of use that the field " Value Min " here over.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Color of the vat

Specify the color of the vat.

Color of the tonnage

Specify the color of the tonnage associated to the vat. It is about the color indicating the level of the vat.

Color of bottom sizes up

Specify the color of bottom of the tonnage associated to the vat.

Option " To display the scale "

If this option is checked, a stepped up scale will be visible to quoted of the tonnage associated to the vat.

Scale Min

This parameter indicates the value of the lower boundary-mark of the scale stepped up of the tonnage associated to the vat. This constant parameter can be of integer type or decimal

Example : 10
0,5

Scale Max

This parameter indicates the value of the lower boundary-mark of the scale stepped up of the tonnage associated to the vat. This constant parameter can be of integer type or decimal.

Example : 100
10,05

Interval graduating

This parameter indicates the value separating two graduating. This constant parameter can be integer or decimal.

Example: 10
1,5

Number of decimals

The graduating can be displayed with decimals. This parameter indicates the number of decimals displayed. This parameter is of integer type.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object ferments will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

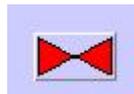
« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

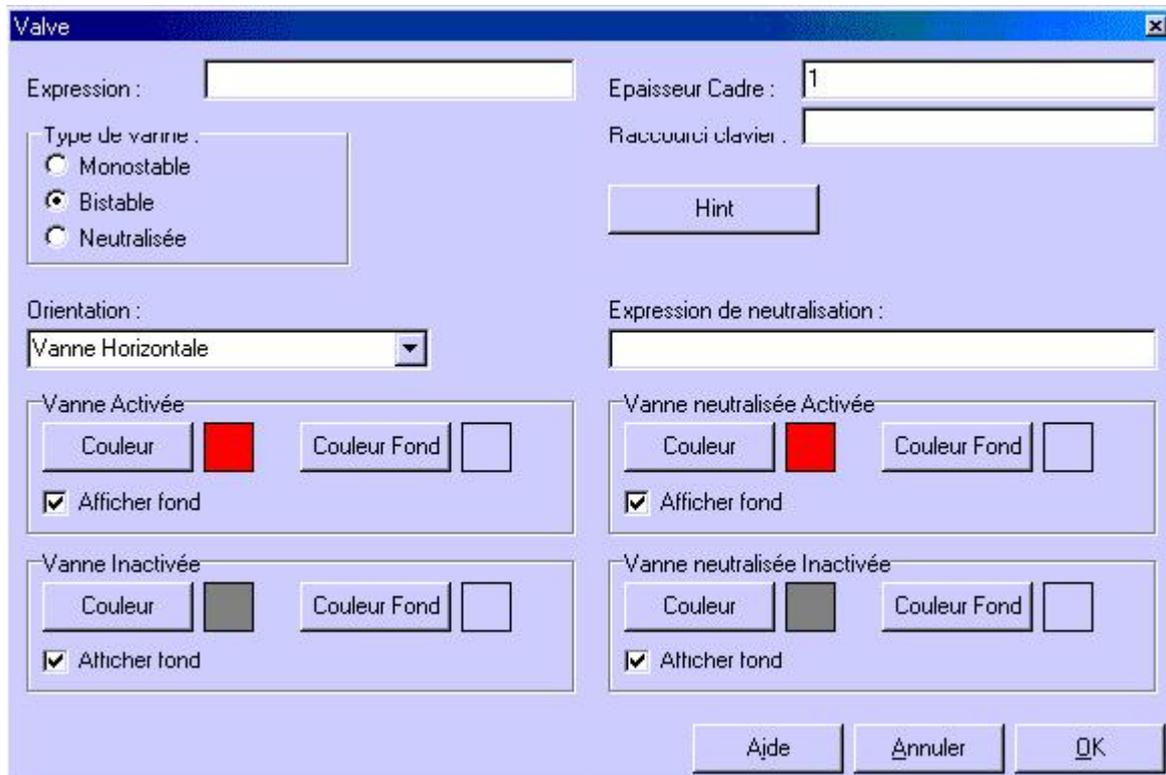
3.2.6.14.2 Floodgate

The object winnows permits to display a floodgate on a supervisory form. This floodgate can control a variable PLC of boolean type. It can neutralize besides by a variable PLC of boolean type.

Example of floodgate :



Dialog box " Floodgate " :



The fields are the next one :

Expression

This window can contain the name of a boolean type variable PLC. If the variable changes state, the floodgate changes appearance consequently. In the same way when the user presses the électrovanne, the variable PLC

is forced to 1.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Type of floodgate (Monostable / Bistable / Neutralized)

This option permits to indicate if the électrovanne is monostable or bistable. If it is monostable, it comes back when one stops pressing it. If it is bistable, it is necessary to click two times to loosen it.

If the électrovanne neutralized, no action is over possible, it serves to the visualization then solely (This neutralization is permanent: to see farther for a temporary neutralization).

Thickness centers

The thickness corresponds to the width of the setting in pixels. More this value is big, more the setting of the électrovanne seems thick.

Shortened keyboard

This option permits to specify a key whose use permits to activate the APIGRAF button directly. The values accepted in this field are :

Letters of HAS to Z, the capitals and the lower-case letters not being differentiated.
The numbers of 0 to 9

Attention to use joined input windows and alphanumeric type abridgment key on buttons! !. Indeed so at the input, one hits the number 9 and that in the same page is a button with " 9 " for abridgment, the button will be engaged.

The keys of functions of F2 (The F1 key being reserved for aide Windows®) in F12
INSERT, SUPPR, ESCAPE, BACK

The combinations of keys with ALT: ALT TO, ALT B, ALT C,.., ALT Z, ALT O. ALT 9

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Vertical Horizontal / Floodgate floodgate

Permits to define the orientation of the drawing of the floodgate.

Zones Floodgate Activated / Deactivated Floodgate

These two zones permit to define the colors and the type of display depending on whether the floodgate is inactive (variable PLC to 0) or active (variable PLC to 1) in the case or the floodgate didn't neutralize.

The colors of drawing and bottom can be defined there as well as the necessity to display or no the bottom.

Expression of neutralization

This window can contain the name of a boolean type variable PLC. If the variable is equal to 1, the floodgate cannot be manipulated more by the user. The floodgate can be activated again by the user.

Example : SA20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA20

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Zones Floodgate Neutralized Activated / Neutralized Floodgate Deactivated

These two zones permit to define the colors and the type of display depending on whether the floodgate is inactive (variable PLC to 0) or active (variable PLC to 1) in the case or the floodgate neutralized.

A floodgate neutralized if it has been defined like neutralized to the level of the type of floodgate or if the variable of neutralization seized in the expression of neutralization is equal to 1.

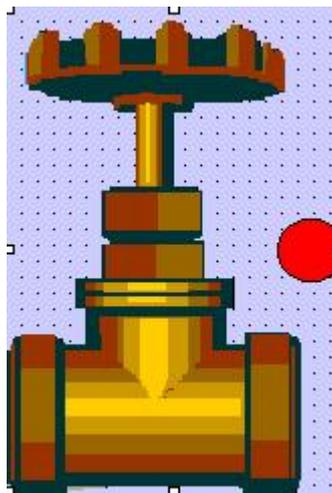
The colors of drawing and bottom can be defined there as well as the necessity to display or no the bottom.

3.2.6.14.3 Floodgate bitmap

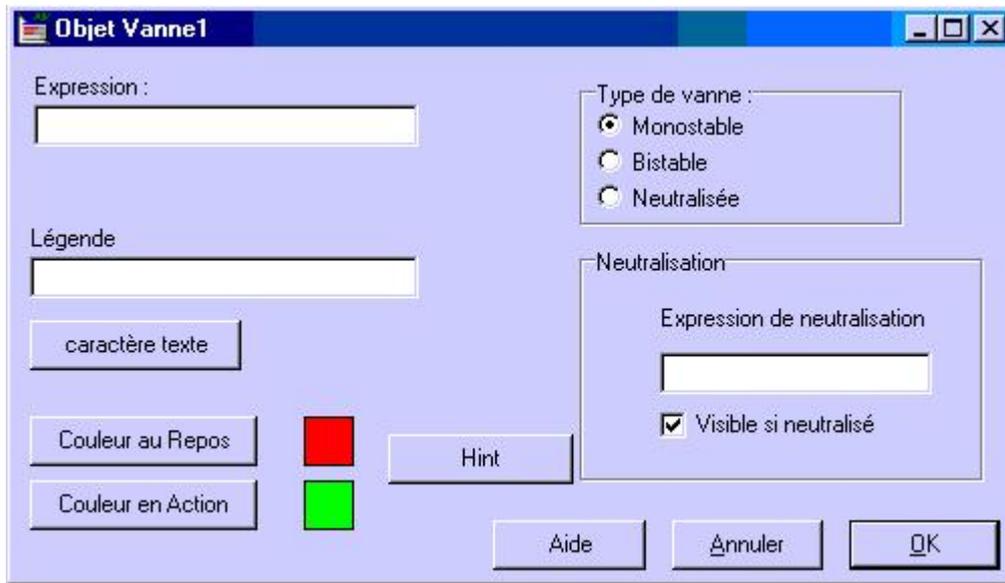
The object winnows bitmap permits to display a floodgate on a supervisory form. The seer of right of this one indicates if she is manipulated or no (green in action and red to rest by default).

This floodgate can control a variable PLC of boolean type. It can neutralize besides by a variable PLC of boolean type.

Example of floodgate :



Dialog box " Floodgate " :



The fields are the next one :

Expression

This window can contain the name of a boolean type variable PLC. If the variable changes state, the floodgate changes appearance consequently. In the same way when the user presses the électrovanne, the variable PLC is forced to 1.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Type of floodgate (Monostable / Bistable / Neutralized)

This option permits to indicate if the électrovanne is monostable or bistable. If it is monostable, it comes back when one stops pressing it. If it is bistable, it is necessary to click two times to loosen it.

If the électrovanne neutralized, no action is over possible, it serves to the visualization then solely (This neutralization is permanent: to see farther for a temporary neutralization).

Color to Rest

Specify the seer's color (on the right of the floodgate) to rest (variable = 0 or False).

Color in Action

Specify the seer's color (on the right of the floodgate) in action (variable = 1 or True).

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

Expression of neutralization

This window can contain the name of a boolean type variable PLC. If the variable is equal to 1, the floodgate cannot be manipulated more by the user. The floodgate can be activated again by the user.

Example : SA20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA20

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Option " visible if neutralized "

If this option is not checked, the object will become invisible when it will have neutralized.

3.2.6.15 HOUR / DATE tab objects

3.2.6.15.1 Display Time

This object permits to display the hour on a supervisory form.

Dialog box "Display Hour"



The parameters are :

- **Characters**

Permits to define the police to use to display the hour (Police, size, color)

- **« Hint » Button**

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

- **« Show Hint » Option**

The Hint text will be displayed in execution mode only if this option is checked.

- **« Hint »**

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

- **Transparent bottom**

When this option is checked, the display of the hour is done without bottom

- **Color of bottom**

To specify here the color of bottom of the text

- **Expression of neutralization**

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object "Display Hour" will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

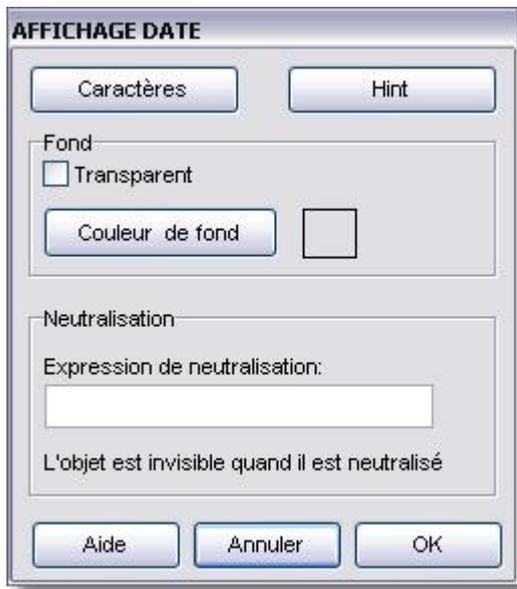
Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.15.2 Display Date

This object permits to display the current date on a supervisory form.

Dialog box "Display Dates"



The parameters are :

- **Characters**

Permits to define the police to use to display the date (Police, size, color)

- **« Hint » Button**

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

- **« Show Hint » Option**

The Hint text will be displayed in execution mode only if this option is checked.

- **« Hint »**

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

- **Transparent bottom**

When this option is checked, the display of the date is done without bottom

- **Color of bottom**

To specify here the color of bottom of the text

- **Expression of neutralization**

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

The object "Display Dates" will be invisible every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

3.2.6.15.3 Calendar

This object displays a calendar and permits to select one date to affect to a variable of type chains.

Dialog box "Calendar"

The parameters are :

- **Expression**

To indicate in this field the name of a the variable of **type chain**.

This variable will be affected by the date selected in the object calendar. If the date contained in the variable changes and is then valid the day current of the calendar will be modified also

Example : C10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !C10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

- **To display the numbers of week**

If this option is checked, the numbers of weeks will be displayed to quoted of the calendar

- **To surround the date of today**

If this option is checked, the dates current (the one of the system) apparaitera surrounded in the calendar

- **To display the date of today**

If this option is checked, the dates current (the one of the system) will be displayed at the bottom of the calendar

- **Display Time / date format OPTIMA PLC**

Specify if the expression is directly a variable of type time, date, date_and_time or time_of_day of optima PLC to the format Dates / Hour.

- **Time type OPTIMA PLC**

Specify the type dates the variable OPTIMA PLC to display. The following choices are possible :
Date, Date_and_time

- **Expression of neutralization**

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The calendar will have neutralized every time that the variable or the expression will be equal to true.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

- **Visible if neutralized**

If this option is checked, the calendar will be visible when it will have neutralized. However, the operator won't be able to act over.

If this option is not checked, the calendar will be invisible when it will have neutralized.

- **Plane rear color**

Specify the color of plane rear of the calendar

- **Color Text**

Specify the color of the texts of the calendar, essentially the dates

- **Color melts title**

Specify the color of bottom of the title in top of the calendar

- **Color titles**

Specify the color of the title in top of the calendar

- **Color dates out month**

Specify the color of the dates that is not in the displayed month.

« **Hint** » **Button**

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.15.4 Date / Hour input box

[Window of Date / Hour \(Details\) input](#)

3.2.6.16 ALARMS tab objects

3.2.6.16.1 Alarm history

This object permits to display the historic of the alarms. The alarms are managed by an OPTIM'ALARM module that can be local (on the same station) or distant.

The object of "Historic supervisory of the alarms" connects (via TCP / IP) on the OPTIM'ALARM module designated by its IP address (for the local mode: 127.0.0.1 or LOCAL_HOST).

This object displays the list of the alarms then.

Date/Heure	Type	Libellé	Priorité
 08/02/2007 13:52:19	104	Porte coupe feu principale type 1	1
 08/02/2007 13:52:24	103	GGGGGGGGG	2

It is then possible to make a double-click on the alarms in order to acquit them under the condition to have the necessary rights if the users management is activated. At the time of the double-click, the following window opens up :



The user can acquit the alarm then. If the users management is activated and that it doesn't have any necessary rights, the system asks him to authenticate himself.

Box of "Historic dialog of the alarms" :



The parameters are :

- **IP address of the OPTIM'ALARM module:**

To indicate in this field addresses it IP of the station or be the OPTIM'ALARM module that centralizes the alarms. If the OPTIM'ALARM module is on the same station, to enter the value 127.0.0.1 for the IP address (or Local_Host).

3.2.6.17 RECORDING tab objects

3.2.6.17.1 Recording of data

The tape recorder of data is useful when the user must stock values of data little with a frequency of acquirement elevated: (1 acquirement all 100ms at several o'clock)

The registration and the storage of the data makes itself entirely with the help of the supervisor who reads prompt manner the data merely in the PLC.

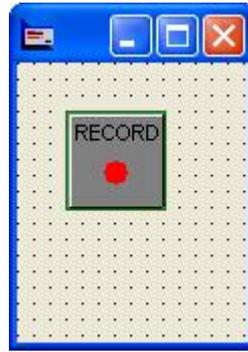
The thus recorded data can be visualized as curves in the object following" Curve Tape recorder that is in the same tab of objects.

The tape recorder can stock until 8 tables and therefore to record 8 different values. It also records automatically the values of time to which every registration has been done.

The data can be recorded cyclically in a file so that the tables are protected in case of resurgence of the supervisory.

The recording object only appears in phase of conception of the supervisory forms and disappears at the time of

the execution. It has the following aspect :



Box of "Recording dialog of data"

ENREGISTREUR

Nom :

Nombre de points :

Enregistrement

cyclique

sur front montant expression

sur front descendant expression

sur front expression

Période : Valeur max : 50000

Base de temps période

milliseconde

seconde

minute

heure

Expression de déclenchement :

Num	Nom du tableau	Expression	Actif
1	TAB0		Non
2	TAB1		Non
3	TAB2		Non
4	TAB3		Non
5	TAB4		Non
6	TAB5		Non
7	TAB6		Non
8	TAB7		Non

Buffer cyclique

Nombre de valeurs effacées en début de tableau quand il est plein :

Enregistrement des données dans un fichier

Nom du fichier :

Temps entre deux enregistrements : Minutes

The parameters are :

- **Name**

Indicate the name of the tape recorder. This name will permit to identify in the curve associated the tape recorder to visualize.

- **Number of points**

Number of points maximal of every table of data of the tape recorder

- **Save**

Permits to select the save mode : cyclic, on rising edge expression, on falling edge expression, on edge expression. The edge indicates a change of state of the boolean expression.

- **Period**

If the selected save mode is cyclic, to enter here the value of the interval between every measure. To specify the units of the interval underneath: millisecond, second, minute, hour.

- **Expression of starting point**

If the selected save mode is: on rising edge expression, on falling edge expression or on edge expression: To enter here the expression that is going to trigger the registrations. This expression must be boolean.

- **List of the tables**

This list shows the 8 tables of registration. Only some tables can be active. It is also possible to reappoint the tables. The names by default being: TAB0, TAB1, TAB7

- **Button to "Modify"**

To parameterize a table of registration (or way of registration), to select the table to parameterize in the list, this one must appear in surbrillance, then to click on the button "to modify." The following paramètre window opens up then :

The image shows a dialog box titled "ENREGISTREUR - TABLEAU". It has a light beige background and a blue title bar. Inside the dialog, there are three main input areas: a text box labeled "Nom du tableau :" containing the text "TAB0", a text box labeled "Expression à enregistrer :" which is currently empty, and a checkbox labeled "Enregistrement tableau actif" which is currently unchecked. At the bottom right of the dialog, there are two buttons: "Annuler" and "OK".

To inform (optional) the name of the table

To inform the expression to record (numeric value)

To check the box "Registration active table" to make efficient the registration in the table.

- **Option "cyclic Buffer"**

If this option is checked, once the table will be filled, a certain number of values will be erased in beginning of table to permit to add new values at the end of table.

- **Number of unobtrusive values in beginning of table when it is full**

If the option "cyclic Buffer" is checked, this parameter specifies the number of values to erase in beginning of table when it is full to permit to add new values at the end of table.

- **Registration of the data in a file**

If this option is checked, all data will be recorded in a file.

- **Name of the file**

To specify the name of the file that will contain all data.

- **Time between two registrations**

Permits to define the interval of time between two registration of files.

3.2.6.17.2 Curve Tape recorder

This object permits to visualize the data coming from the "Recording object of data." This object is on the same tab of object.

Box of "Recording dialog of data" - "General Tab"

GRAPHE ENREGISTREUR

Général Axes et Echelles Titres Légendes Fond Neutralisation Seuils Impression Zoom 3D Autres

Nom de l'enregistreur :

Tableaux de mesures sur axe Y

Nom	Unité

Ajouter Supprimer

Affichage Temps

Afficher temps sur l'axe des X

Format des dates/temps :

dd/mm/yy "hh:nn" ▾

Hint

Type d'affichage

Traits
 Histogramme

Type de barre

Rectangle
 Rectangle dégradé
 Pyramide
 Pyramide inversée
 Flèche

Echelle en Y

Axe Y1
 Axe Y2

Options d'affichage

Couleur de la courbe

Aide Annuler OK

The parameters are :

- **Name of the tape recorder**

Indicate the name of the tape recorder to visualize.

- **Tables of measure on Axis there**

Indicate the name of the tape recorder to visualize.

To click on the button "to Add" to add a new curve of the tape recorder to visualize. This parameter corresponds to the ways of the tape recorder: TAB0 (names by default of the tape recorder), TAB1,.. TAB7.



To enter the name of the table of the tape recorder (TAB0,... TAB7 or other if the name has been modified by default).

- **Display time**

To check this option so that the curve indicates in abscissas the different horodatages of the acquisitions.

- **Format of dates / time**

to select here a format of display of the horodatage

Type of display

Permits to select the type of display for the curve selected in the list of the measure tables. There are two possibilities

- Milked
- Histogram

Bar type

This option is only accessible if the type of graph is a histogram. It permits to select the type of bar of the graph.

Scale in THERE

There are 2 available vertical scales in the graph, an on the left and (Y1 (Y2)) an on the right. This field permits to specify what vertical scale will be used for the display of the curve.

Color of the curve

This button permits to specify the color of the curve selected in the list of the measure tables.

« **Hint** » **Button**

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



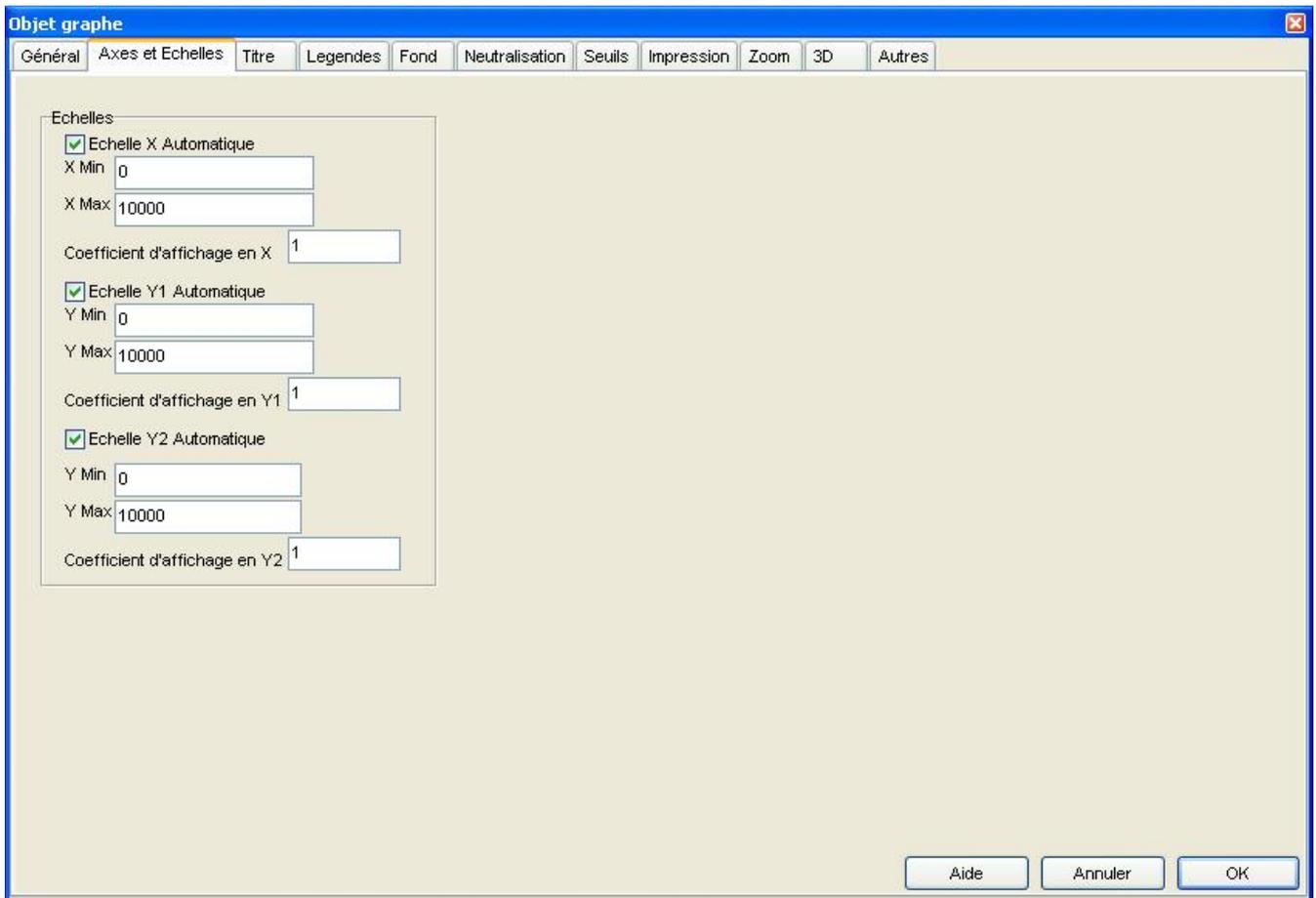
The parameters are :

« **Show Hint** » **Option**

The Hint text will be displayed in execution mode only if this option is checked.

« **Hint** »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

Dialog box graph - Tab "Axes and Scales"**Scales**

Permits to specify the scales of the three axes following X, Y1 and Y2. The Y2 axis can be used in the case or several curves don't have the same units.

There are two fashions of scales, the automatic mode and the manual mode.

In the automatic mode, it is necessary to check the box " X scale, Y1 or Y2 Automatique " following the concerned axis. In this case, the values Min and Max of the scale will be adjusted to the values of the tables and the fields Min and corresponding Max to the scale don't have need to be informed.

If the box " Automatic scale " is not checked. The values min and max of the scale will be the values appearing in the fields Min and Max of the scale.

Coefficients of display

It is possible to specify for each of the three available scales a coefficient of display. This coefficient defines the report between the displayed value and the real value in the table.

Option "Scale X Automatique"

To check this option so that the extent of the scale of the X automatically adapts to the values of the curves. If this option is not checked, it is possible to give the XMin values and XMaxes of the scale of the X.

Coefficient of display in X

Its value is by default 1. This value is multiplied to the values of the measure table in X.

Option "Scale Y1 Automatique"

To check this option so that the extent of the scale of the Y1 automatically adapts (Scale of left) to the values of the curves. If this option is not checked, it is possible to give the Y1Min values and Y1Maxes of the Y1 scale.

Coefficient of display in THERE

Its value is by default 1. This value is multiplied to the values of the measure tables positioned in relation to the Y1 (axis of left) axis.

Option "Scale Y2 Automatique"

To check this option so that the extent of the scale of the Y2 automatically adapts (Scale of right) to the values of

the curves. If this option is not checked, it is possible to give the Y2Min values and Y2Maxes of the Y2 scale.

Coefficient of display in THERE

Its value is by default 1. This value is multiplied to the values of the measure tables positioned in relation to the Y2 (axis of right) axis.

Dialog box graph - Tab "Titles"

The screenshot shows a dialog box titled 'Objet graphe' with a blue title bar and a close button in the top right corner. The dialog has a tabbed interface with the following tabs: 'Général', 'Axes et Echelles', 'Titre' (selected), 'Legendes', 'Fond', 'Neutralisation', 'Seuils', 'Impression', 'Zoom', '3D', and 'Autres'. The 'Titre' tab contains five text input fields:

- 'Titre du graphe': A text input field for the main title.
- 'Expression à ajouter au titre': A text input field for an expression to be added to the title.
- 'Titre Axe X': A text input field for the X-axis title.
- 'Titre Axe Y1': A text input field for the Y1-axis title.
- 'Titre Axe Y2': A text input field for the Y2-axis title.

At the bottom right of the dialog, there are three buttons: 'Aide', 'Annuler', and 'OK'.

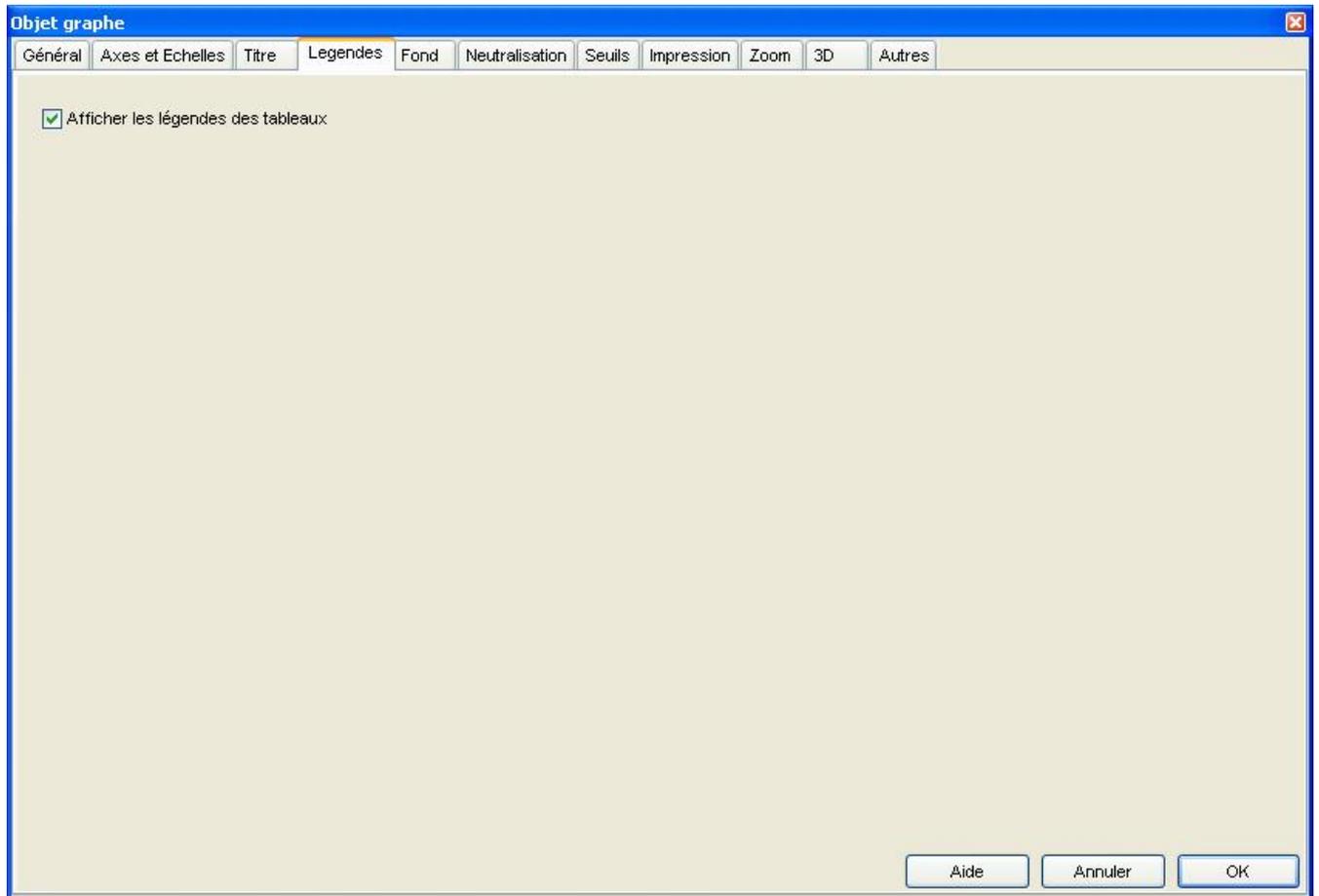
Titles

A general title can be given to the curve as well as legends for each of the three available axes.

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

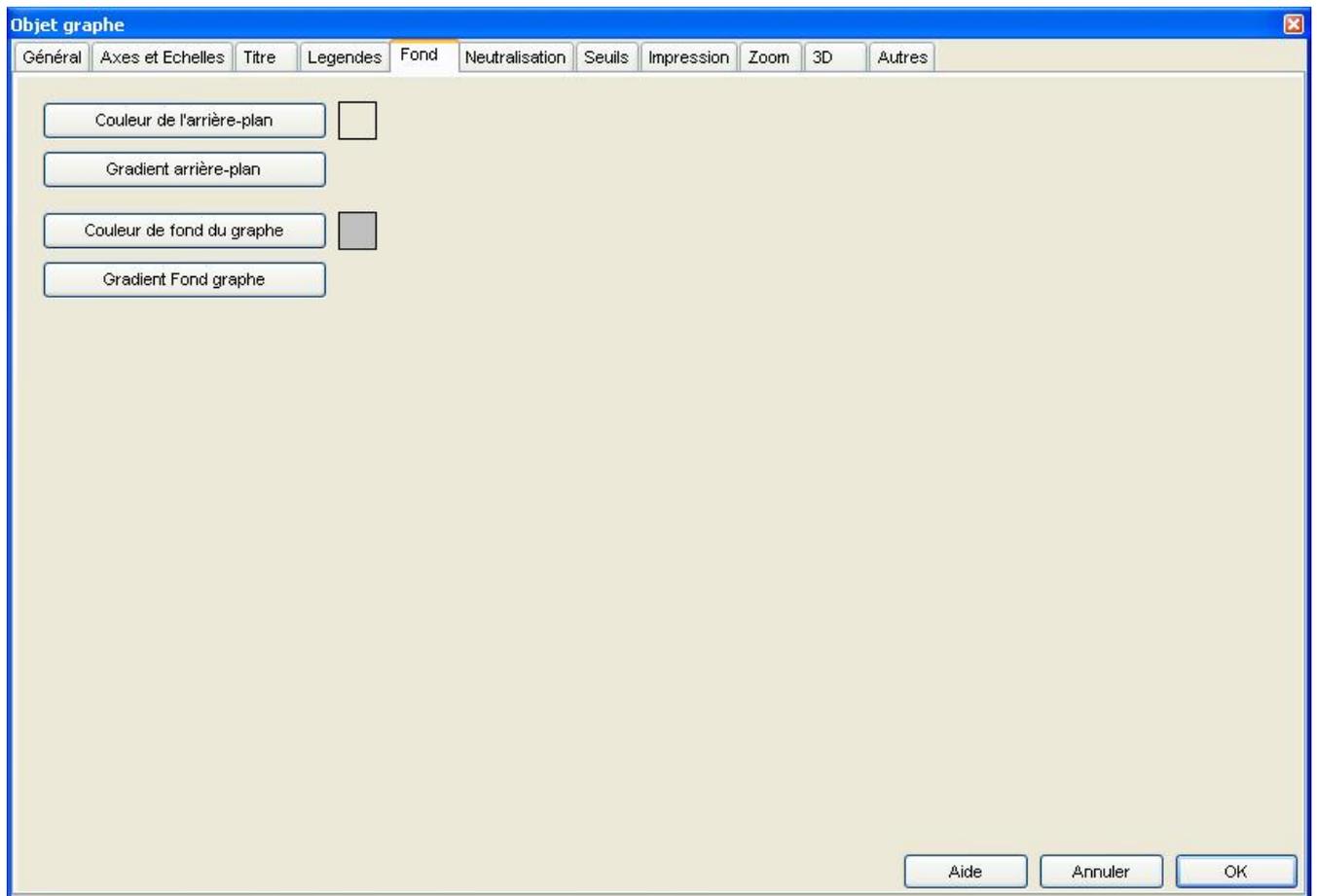
One can also specify an expression to add to the title. For example a serial number contained in a register: R100.

Dialog box graph - Tab "Titles"



To check the option to "Display the legends" to display the names of the samples tables associated to the colors of the corresponding curves.

Dialog box graph - Tab "Melts"

**Color of the rectangle of the graph**

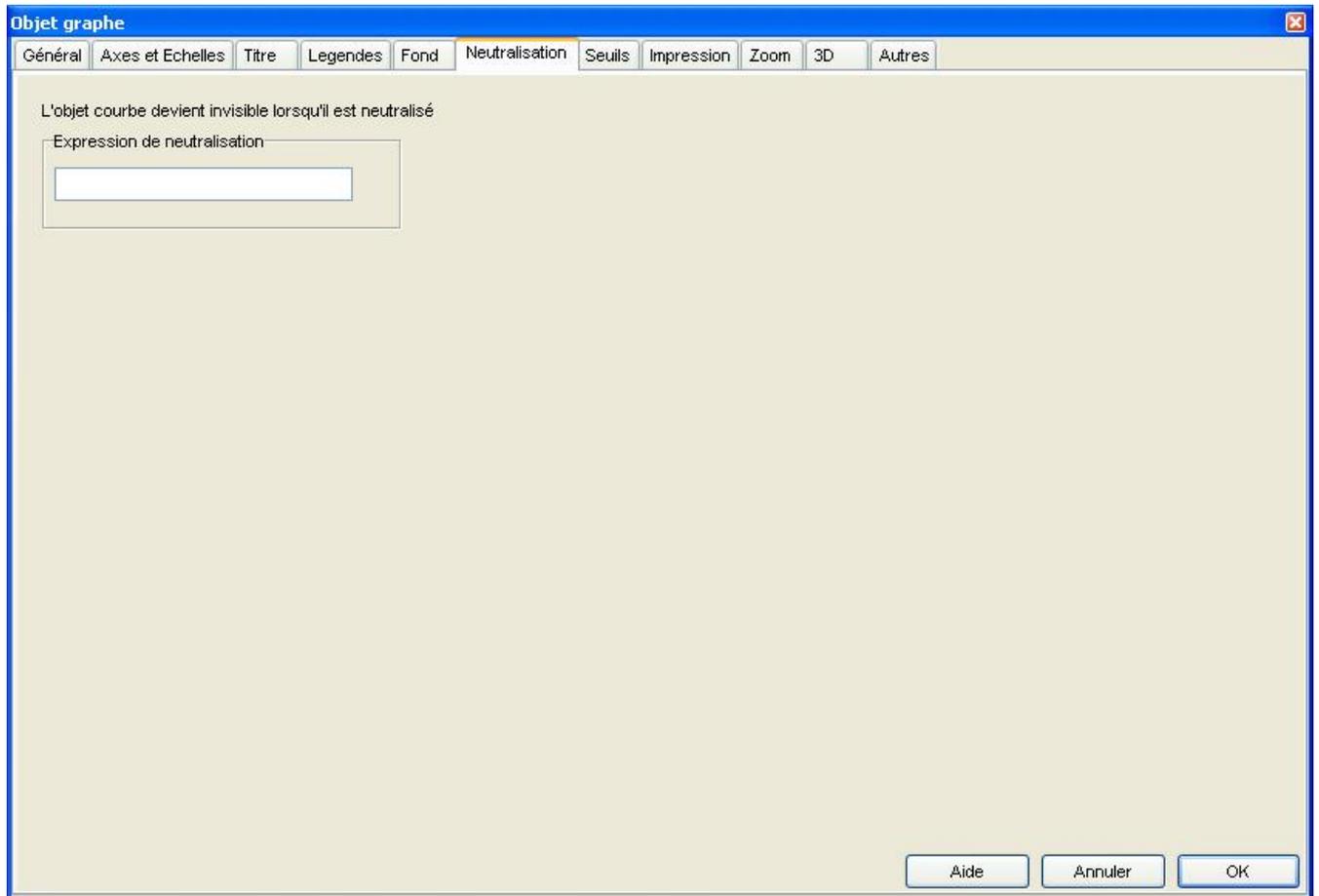
This button permits to choose the color of plane rear of the graph.

Color of the plane rear

This button permits to choose the color of the contour of the object graph.

Some buttons gradients also permit to define of damaged them of color in bottom of the object and behind plane of the curve.

Dialog box graph - Tab "Neutralization"



Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type). The object bends will be invisible every time that the variable or the expression will be equal to true.

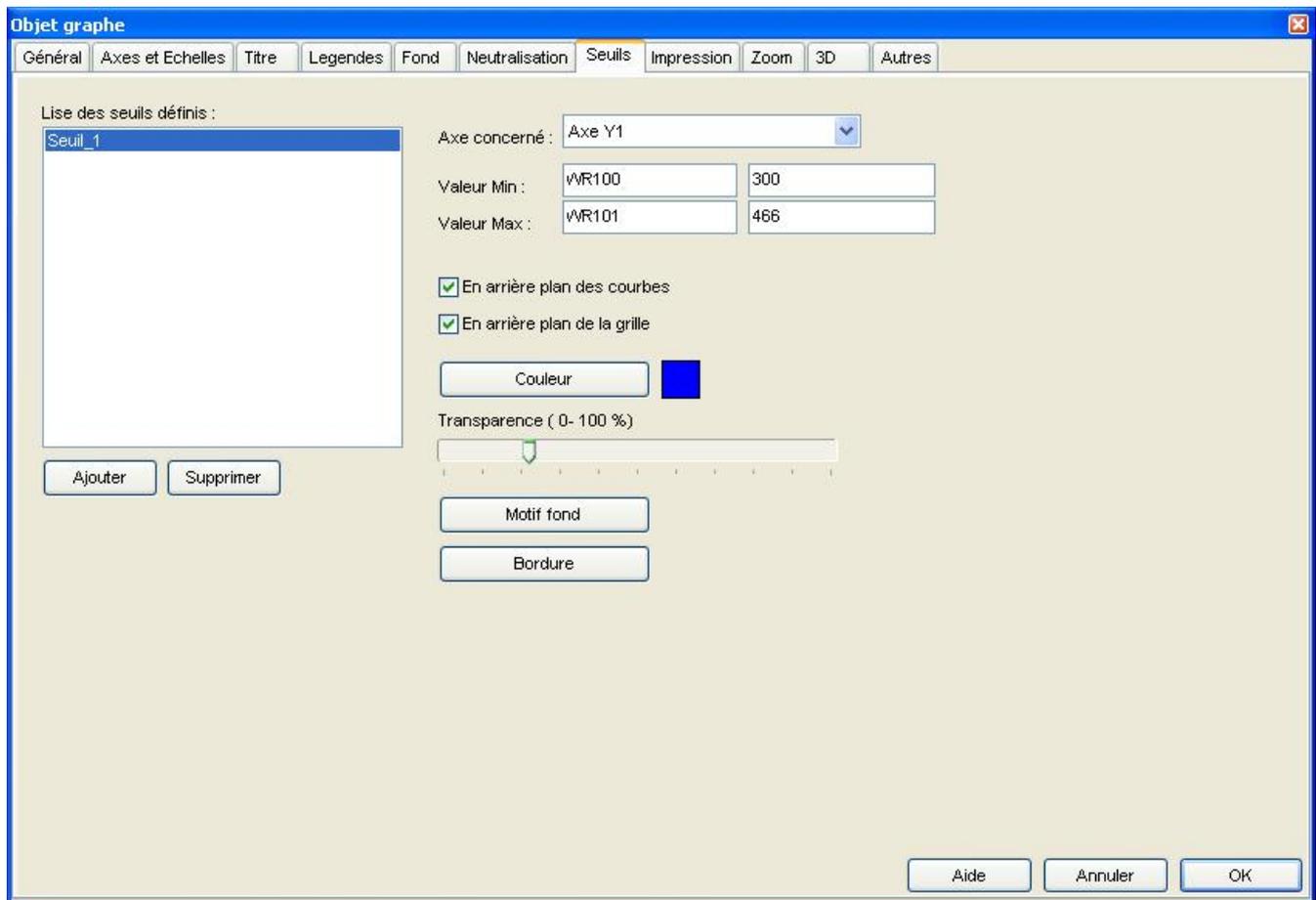
Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Dialog box graph - Tab "Doorsteps"



The doorsteps are the colored zones that show off on the curves and that permit to visualize some zones limits in general. Some doorsteps can be added or can be suppressed in the unwinding list with the help of the buttons "to Add" and "to Suppress." Once an added doorstep, it is possible to parameterize its aspect :

Axis concerned :

Defines the axis to which relates the doorstep, a doorstep can be therefore horizontal if it referred in Y1 or Y2 or vertical if it referred to the X axis.

Value Min and Value Max :

These values define the extent of the zone doorstep, one can enter the constant numeric values or the numeric expressions.

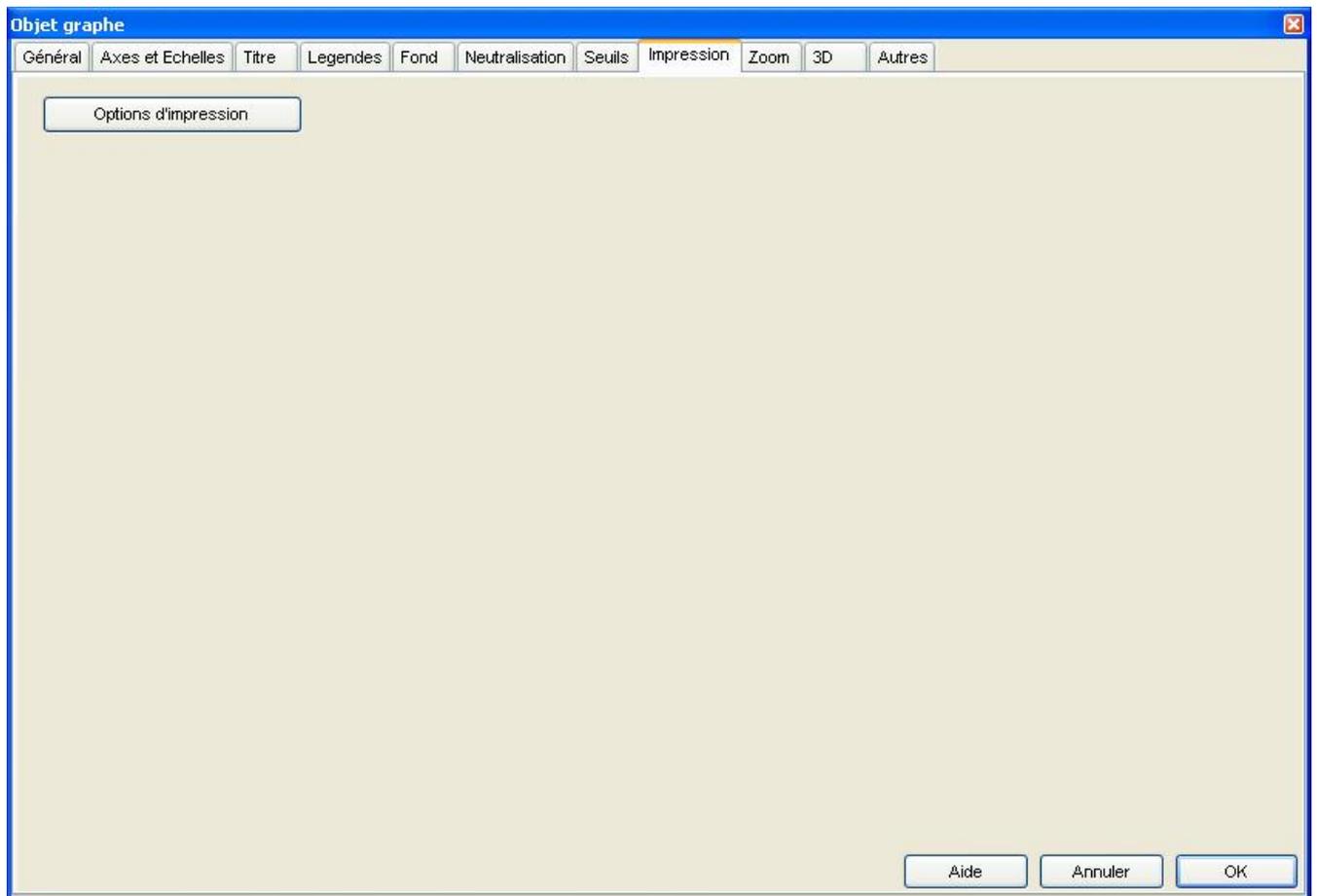
Example : R10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R10

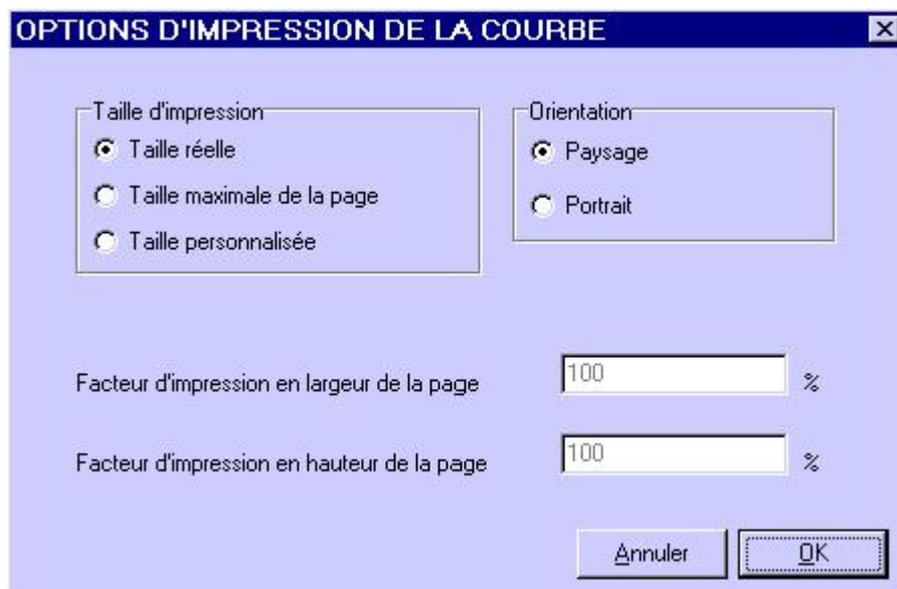
The appearance of the zone doorstep can also be redefined: the transparency, the motive of the bottom, the color,...

Dialog box graph - Tab "printing"



Options of printing

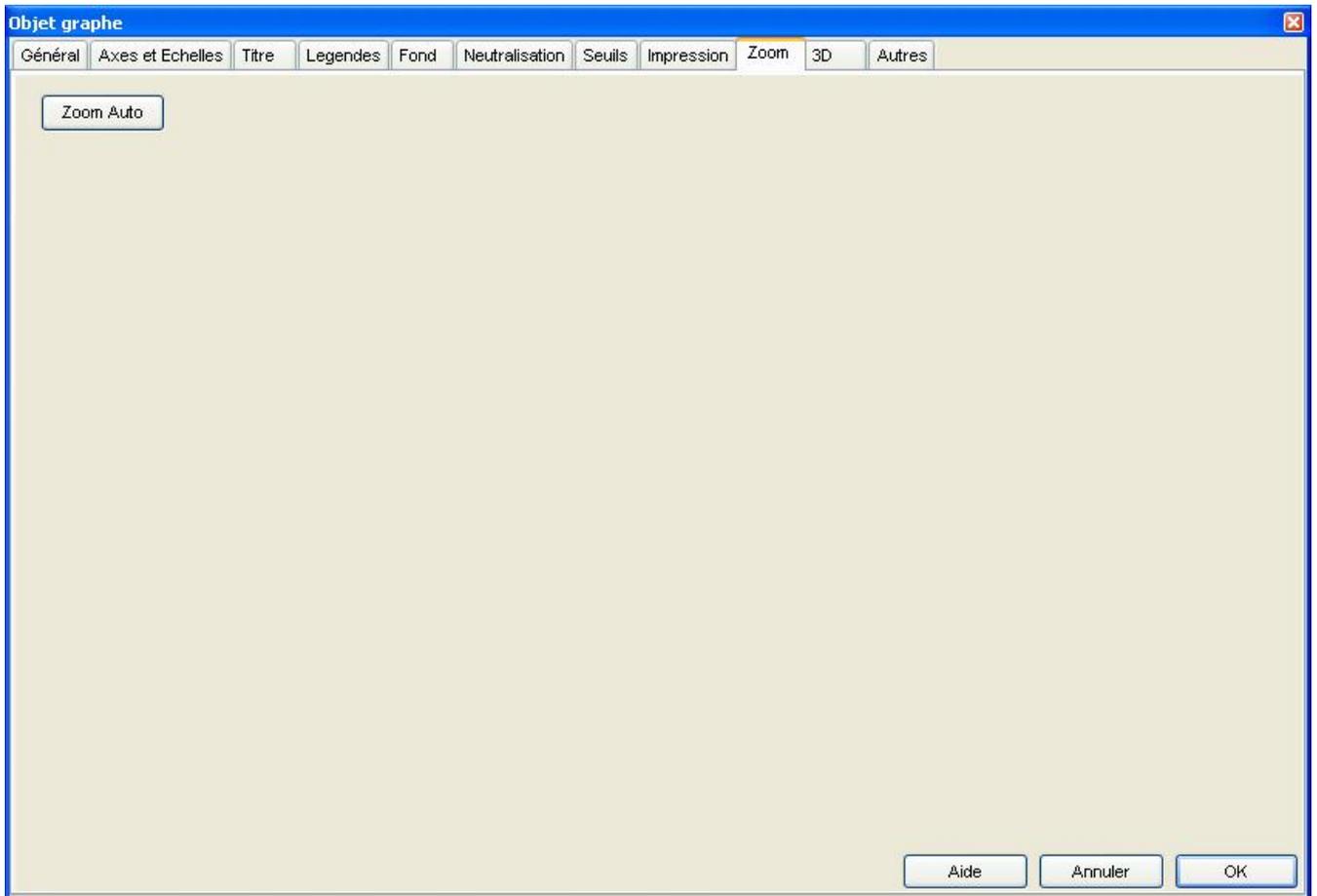
The curves can be printed at execution. It is necessary for it to double-click on the curve to print at execution. The options of printing permit to parameterize the printing of the curve :



If the option " Size personalized " is chosen, the factors of printing in width and in height of the page will be taken in account for the printing. These factors will be applied to the size of display to the screen of the curve.

The option " Maximal size of the page " go spread the curve on the whole size of the output paper. In this case the definition of printing is less good.

Dialog box graph - Tab "Zoom"



The button "Zoom" opens the following window :

Zoom Automatique

Expression zoom plus abscisses :

Expression zoom moins abscisses :

Pas du zoom en pixels :

Expression shift plus abscisses :

Expression shift moins abscisses :

Pas du shift en pixels :

Expression reset zoom :

Annuler OK

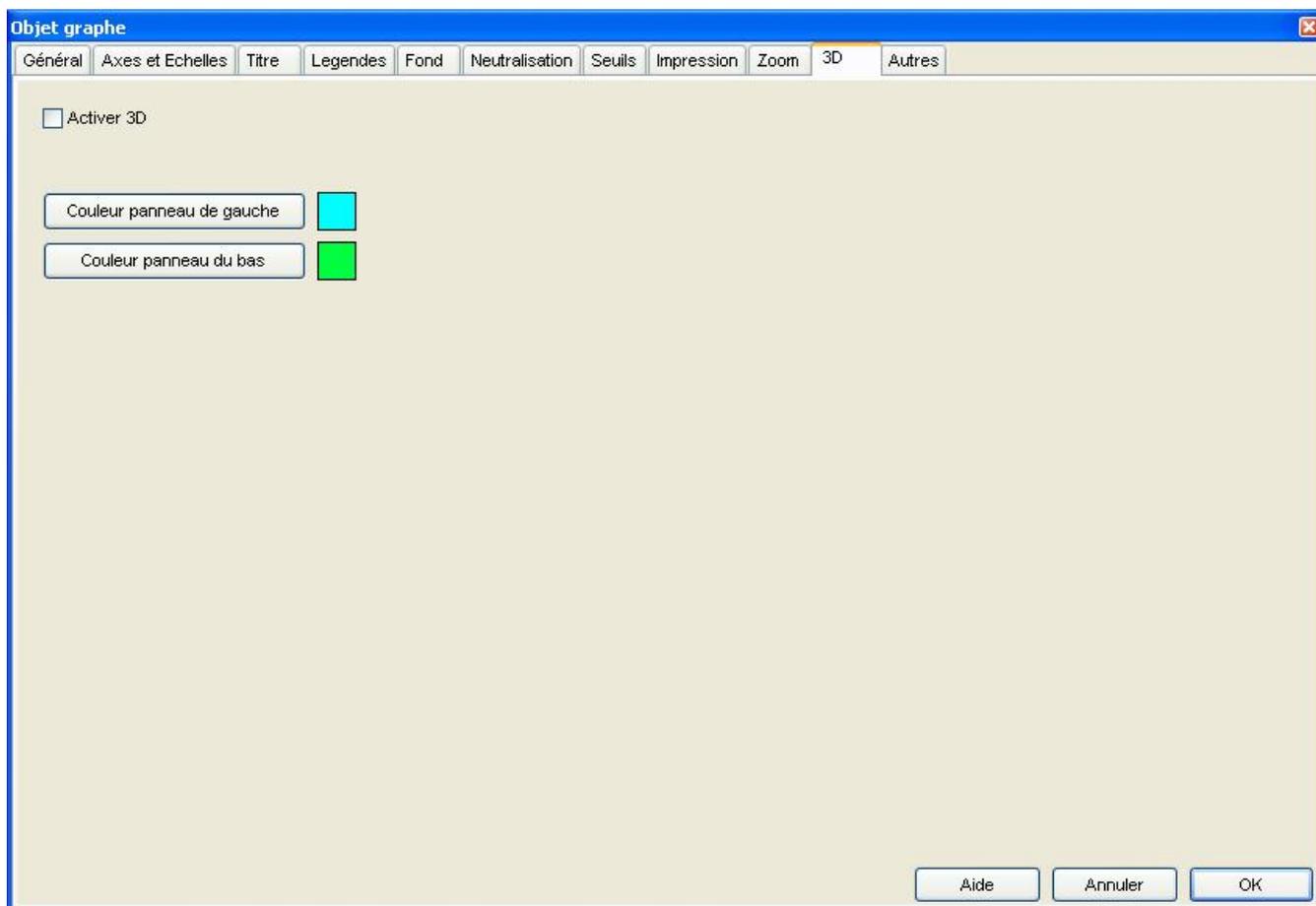
This window permits to associate variables of control of the zooms of the curve.

Example : To enter SA10 in the input box "Expression zoom more abscissas." To define a button then API monostable with the SA10 value. To every pressure of the button, the curve is going to be zoomed in in X with the specified (2 by default) step.

One can make the same thing to control the displacement ("Shift") of the curve.

An expression reset zoom permits to annul all zooms done on the curve.

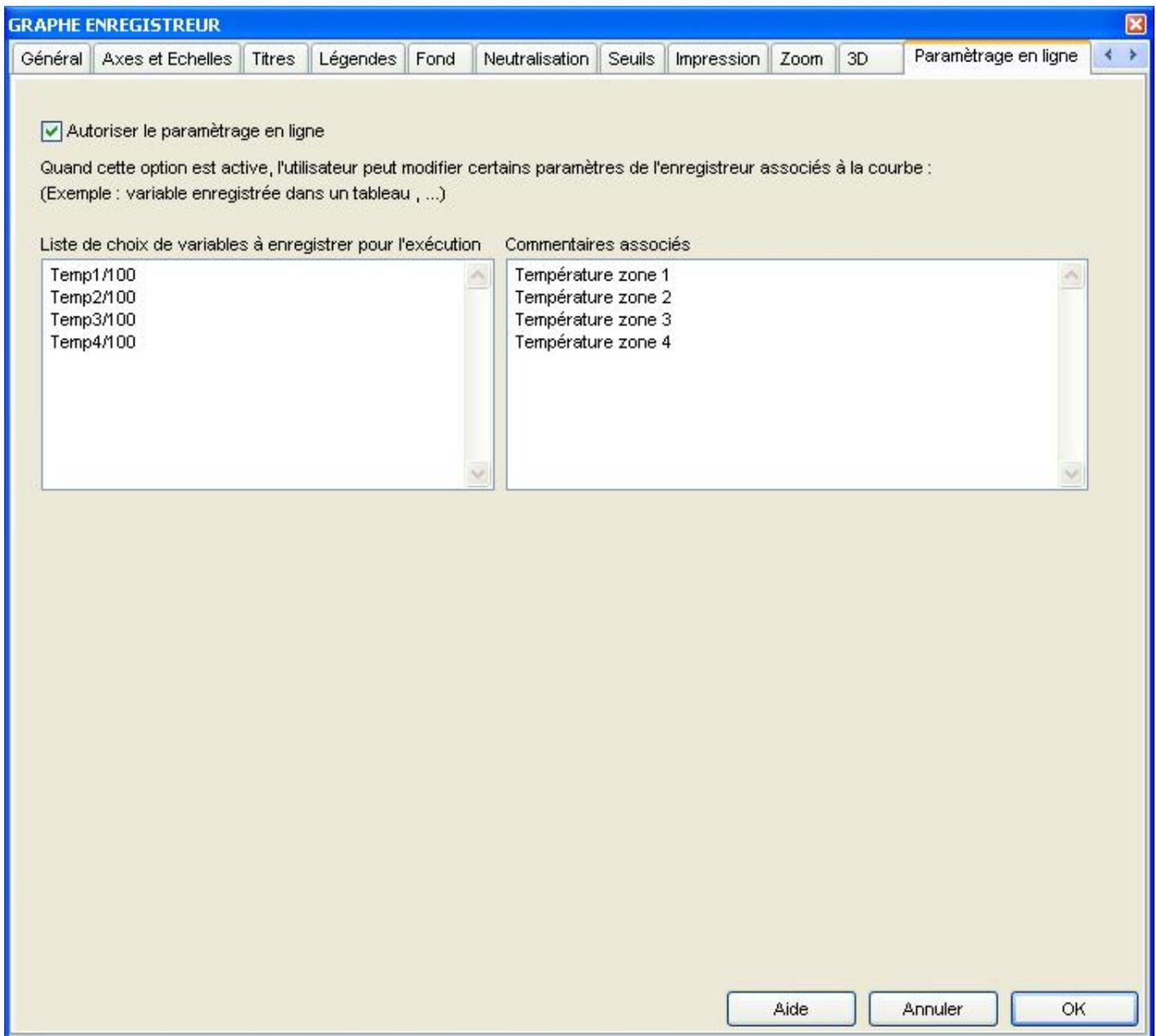
Dialog box graph - Tab "3D"



Option to "Activate 3D"

Permits to display the window of the graph in 3 measurements. One can define the colors of the panels of left and the low created by the visualization 3D then.

Box of "Recording dialog of data" - Tab "on line Paramétrage"



Option to "Allow the on line paramétrage"

This option when it is activated permits to the utilisateur to modify the parameters of the data of the tape recorder associated to the curve in phase of execution of the supervisory.

List of choice of variables to record for the execution

This list is going to define all variables that will be able to be chosen in a list in phase of execution. To enter in the field a variable by line

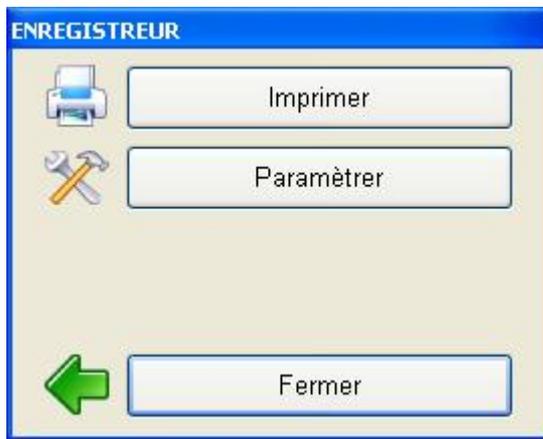
Commentaries partners (optional)

This list permits to associate some commentaries for every variable of the list. These commentaries apparaitront when the user will choose a new variable.

To enter in the field a commentary by line.

The commentary of the first line correspondera to the variable of the ligne1 etc...

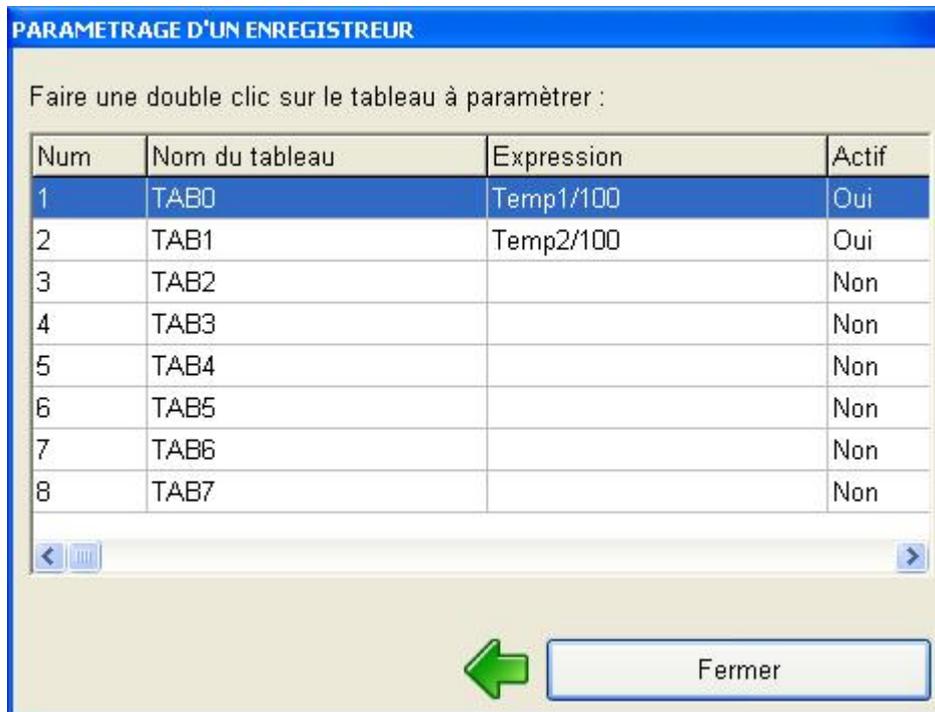
If the option "**To allow the on line paramétrage**" is active, in this case a double click on the centers the curve during the execution of the supervisory will have the effect of to open the window of menu follow :



The button "to print" permits to print the curve by default on the definite printer

The "parameterize" button opens the on line paramètrage window :

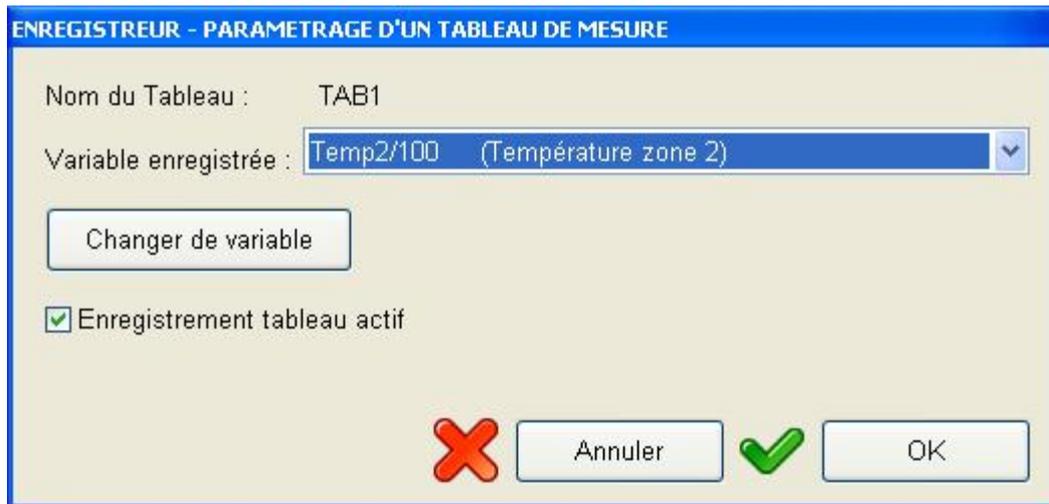
Example :



The user can modify one of the tables of the list by making a double click on this one. The following window opens up :



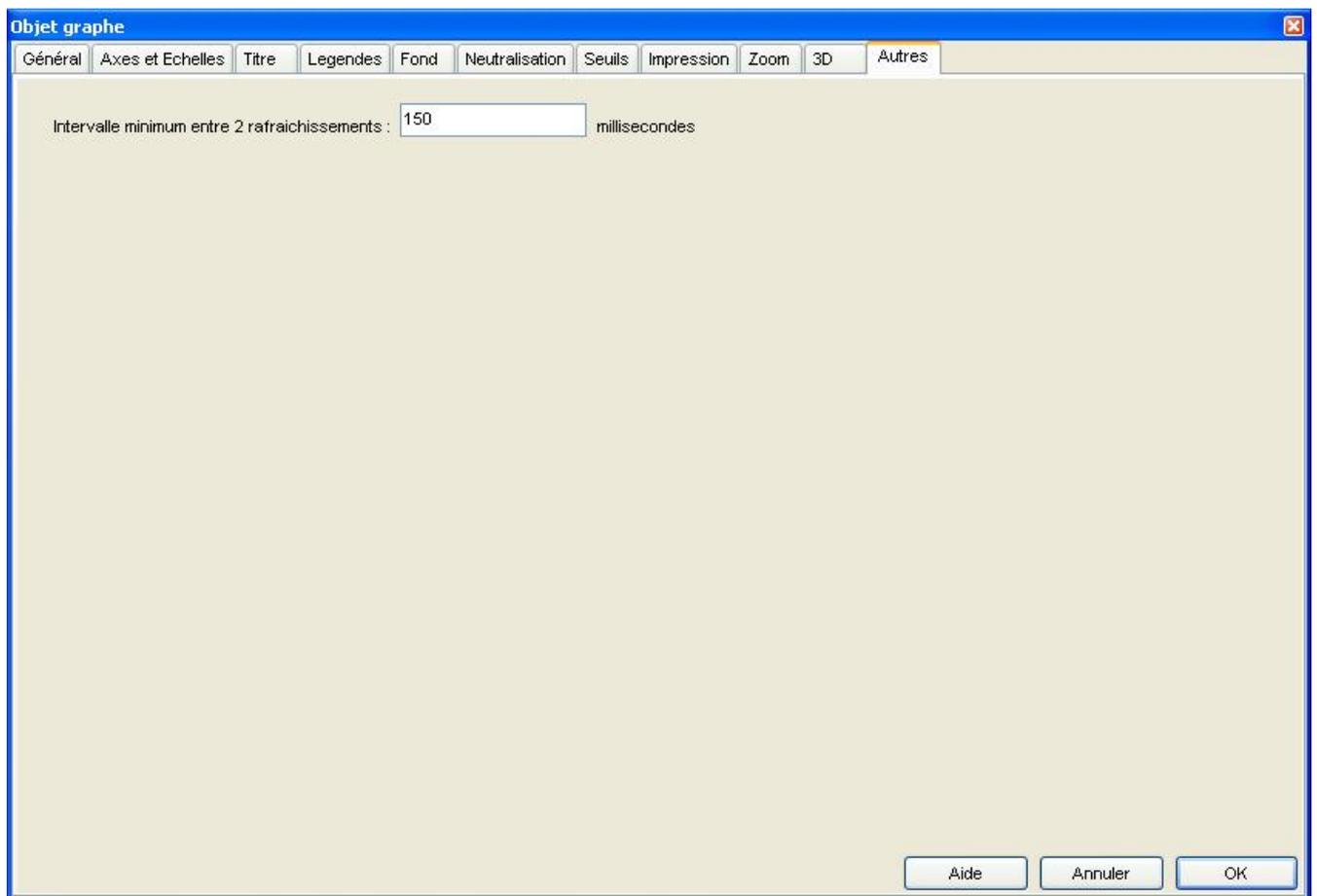
To choose a new variable in the list of the possible variables, to push on the "Variable" button. A Selector to unwinding list of variables appears then on the window :



It is also possible in this window to inhibit or to activate the registration of a table.

Remark : if a modification has been done, all tables of measure of the tape recorder will be handed to 0.

Dialog box graph - Tab "Others"



The complex displays of voluminous curves ask for a lot of resources. One can limit the frequency of rafraichissement of the displays therefore.

3.2.6.18 MULTIMEDIA Tab objects

3.2.6.18.1 Média player

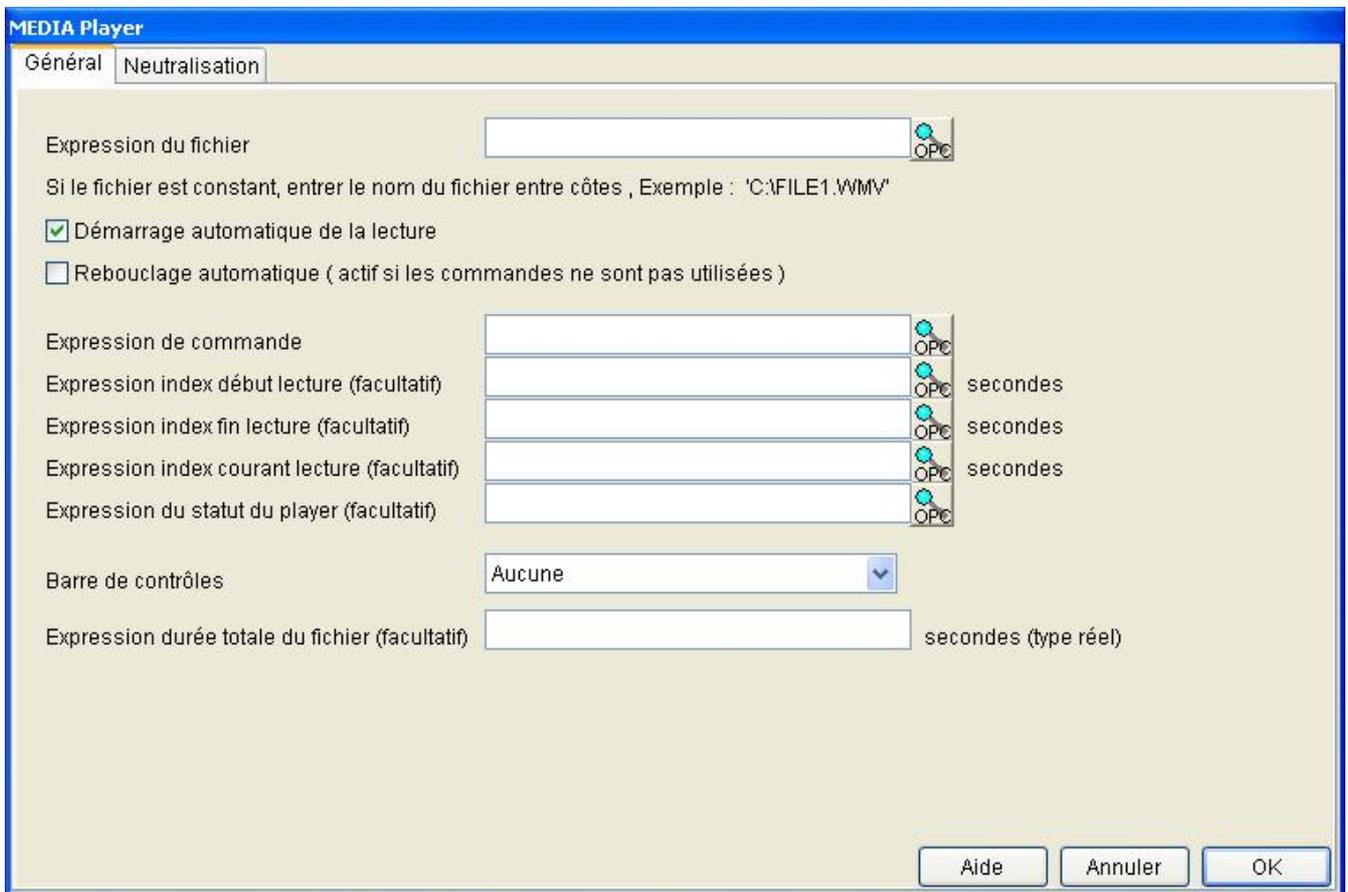
The Media player object can display video files into a supervision form. This object use in fact the Windows media player program.

In some cases, to play special kinds of video, you have to update the Windows media player or to update the codecs used by the windows media player.



The media player in designing mode is a black screen where the videos will be displayed.

Dialog Box Media Player - General :



File expression

It's possible to specify a video file name or a PLC variable of type string which contains the video file name to play.

For a simple video file name, use the single quotes to enter the file name :

Example : 'C:\TEST.AVI'

For a PLC variable, enter the variable name :

Example : C13

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !C13

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

In the case of the variable, the video file name to play is dynamic.

"Auto Play" option

If this option is checked, the video will automatically be started. In the case of the dynamic file name, the video will be restarted each time the file name will be changed.

"Auto Replay" option

If the commands of the media player (See further) are not used, the video file will be replayed automatically each time it will reach the end.

Command Expression (Optional) :

Specify here a PLC variable (numerical integer, example : R10) . The value of the variable will execute some commands on the player :

0	No command
1	Play
2	Stop
3	Hold (pause)
4	Continuous play

Start index play expression (Optional) :

Specify here a PLC variable (numerical type). The value of the variable can be affected with the start index of the video to play (in seconds). The expression can be a constant or a PLC variable.

Examples : 1 , R10,...

If the value of the variable is 1, the play will start the video at 1 second.

Stop index play expression (Optional) :

Specify here a PLC variable (numerical type). The value of the variable can be affected with the stop index of the video to play (in seconds). The expression can be a constant or a PLC variable.

Examples : 1 , R10,...

If the value of the variable is 10, the video will be stopped at 10 seconds.

Current index play expression (Optional) :

Specify here a PLC variable (numerical type). The value of the variable will be automatically be affected with the current index of the video (in seconds)

Media Player Status expression (Optional) :

Specify here a PLC variable (numerical integer type). The value of the variable will be automatically be affected with the status of the media player.

If status value is 3, the current video is being played.

If status value is <=1, 9 or 10, the current video is not played (stopped or ended).

Media player control bar :

Select here the control bar mode: None (with no control bar displayed) or complete.

Current video duration Expression (Optional) :

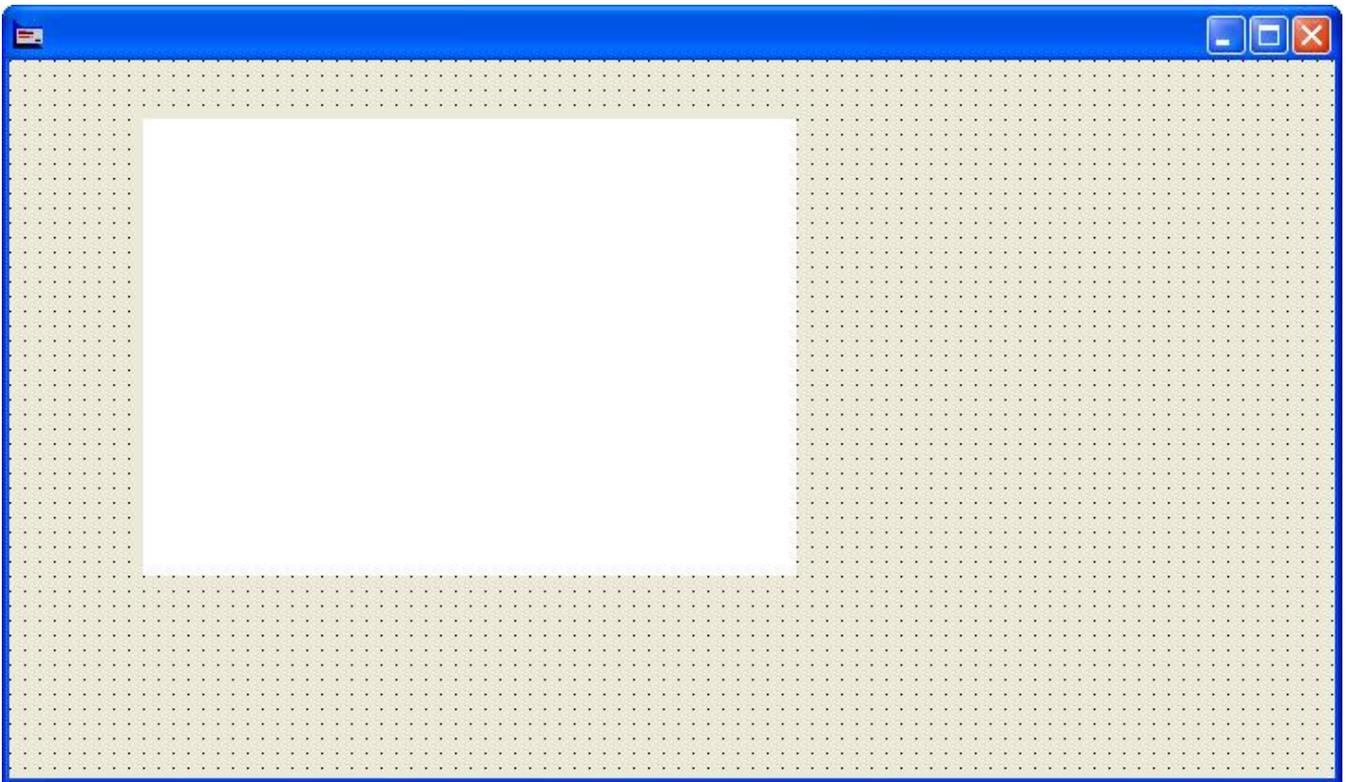
Specify here a PLC variable (numerical type). The value of the variable will be automatically be affected with the duration of the current video (in seconds)

Dialog Box Media Player - Neutralization :

The user can specify a PLC variable (Boolean). When the value of this variable will be 1, the media player will become invisible. It will be displayed back when the value will be 0.

3.2.6.18.2 Flash ShockWave player

The flash player can play shockwave flash files. this object use the Adobe ShockWaveFlash ActiveX. This object can need to download and install the adobe ShockWave Flash Player available for free on internet.



The objet looks like a blank screen where flash files are played

Dialog Box "Flash player" - General Tab :

Objet FLASH

Général Neutralisation

Expression du fichier OPC

Si le fichier est constant, entrer le nom du fichier entre côtes , Exemple : 'C:\FILE1.WMV'

Démarrage automatique de la lecture

Rebouclage automatique (actif si les commandes ne sont pas utilisées)

Expression de commande player (facultatif) OPC

Expression du statut du player (facultatif) OPC

Aide Annuler OK

File Expression

The user can enter here a flash file name or a PLC variable of type string which value is the flash file name to play:

To specify a flash file, enter it with quotes :

Example : 'C:\TEST.SWF'

To Specify a PLC variable, enter the variable name :

Example : C13

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !C13

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

In the case of a variable, the flash file name is dynamic and correspond to the variable value.

"AutoStart playing" option

If this option is checked, the flash file will be replayed automatically. In the case of a variable file name, the file will be played automatically each time the file name will be changed.

"Auto loop" option (available if command mode is not used)

If the command expression is not used (see above) and this option is checked, the flash file will be replayed automatically each time it will end.

Command expression (Optional) :

The user can specify here a PLC variable of integer type (Example : R10,...) which will command the player. The value of the variable will execute specific actions :

0	No Action
1	Play
2	Stop
3	Hold
4	Continuous play

Player Status expression (Optional) :

Enter here a variable with an integer type. The value of the variable will be assigned with the player status value.

If the status is 3, file is playing

If the status is 0, file is stopped or paused

Dialog Box "Flash player" - Neutralization tab :

The user can specify here as the neutralization expression a PLC variable of Boolean type. The player will be invisible when the value of the neutralization expression will be set to 1.

3.2.6.19 OTHERS tab objects

3.2.6.19.1 Recipe

This type of object permits the creation of returns in the supervisory form.

A recipe is a set of values given to a set of variables PLC.

The returns are protected in files RECIPE that have the extension :

*.RECW

An object " RECIPE " results to the execution in a button.

At the time of the execution of the page, a click of the mouse on the button associated to the recipe opens a dialog box in which the values of the recipe of which displayed.

You can enter new values, to recall safeguards of returns, then there to protect the returns, to transfer the data of the recipe toward the data PLC,...

It depends on the possibilities that are parameterized in this object.

Dialog box " recipe " :

Field " List of the variables " :

All variables that you wish to see in the recipe must be enumerated by their name and must be separated by a comma.

Continuations of variables can be defined by indicating the type of variable and the indication of the first element, followed of two points (..) and of the indication of the last element (Xi..j). The visualization of a complement logical of a boolean variable is gotten by making it precede the character \ (Xi).

Example :

The following declaration :

R0, R3.. 5, G3, L40.. 45, SA2, \SA3. .5, A5, DT10, C15

Will permit to visualize the following variables in the recipe :

R0 R3 R4 R5 G3 L40 L41 L42 L43 L44 L45 SA2 \SA3 \SA4 \SA5 A5 DT10 C15

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Shortened keyboard

This option permits to specify a key whose use permits to activate the button RECIPe directly. The values accepted in this field are :

Letters of HAS to Z, the capitals and the lower-case letters not being differentiated.

The numbers of 0 to 9

Attention to use joined input windows and alphanumeric type abridgment key on buttons! !. Indeed so at the input, one hits the number 9 and that in the same page is a button with " 9 " for abridgment, the button will be engaged.

The keys of functions of F2 (The F1 key being reserved for aide Windows®) in F12

INSERT, SUPPR, ESCAPE, BACK

The combinations of keys with ALT: ALT TO, ALT B, ALT C,..., ALT Z, ALT O. ALT 9

Possibilities

The possible choices are :

-All possibilities :

All functions possible of the recipe are accessible (consultations, modification, safeguard, loading, creation)

-To open only

When the object recipe is activated (that is to say when one clicks on the button associated to the recipe), a box containing the names of the existing files appears. The operator can only choose one of them and can validate it. The recipe is transferred then immediately in the variable PLCS.

-To save only

At the time of the activation of the object recipe, the operator can enter the name of a file, in which the variable PLCS of the recipe is protected. If the file already exists, this one will be crushed.

-To create only

At the time of the activation of the object recipe, the operator can enter the name of a file, in which the variable PLCS of the recipe is protected. The file must not exist otherwise the operation won't be possible.

-To put up to date only

When the object recipe is activated, the variable PLCS of the recipe are protected in the file specified in the field " Name of the file by default " (This field only appears if the option " to put up to date only is selected "). There is not choice of file therefore for the safeguard.

Type of button

This field permits to choose between an oblong button or a round button (or elliptic).

Directory files

Permits to specify the under directory in which is or will be the files returns.

If the field is empty, the files returns will be in the directory of the project.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Drawn

The wording is the name that you give to your recipe.

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Thickness of the setting

Specify the thickness of the setting of the button RECIPE.

Texte of the button

Specify the text that appears on the button RECIPE.

It is possible to bring a reference to a parameter in this field (example: {1} to see Section " Call of a supervisory form with passage of parameters ")

Color of the Bottom

Permits to modify the color of bottom of the button RECIPE.

Button " Characters "

Permits to specify the font used by the text of the button RECIPE.

Button " Picture of bottom "

Permits to specify a Picture Bitmap that will appear in bottom on the button of the RECIPE.

Button " To suppress Picture "

Permits to suppress the picture of bottom associated to the object recipe.

Option " To adjust the size of the picture to the one of the object "

If this option is checked, the picture will appear stretched on the whole surface of the button of the RECIPE.

Option " Automatic transparency "

When this option is activated, the color of bottom of the determined picture will automatically be transparent.

Group of fields " Neutralization "

These fields concern the neutralization of the button associated to the recipe. The button, if it neutralized, won't be more actionable. All actions on the recipe won't be therefore feasible.

Expression of neutralization

Permits to specify a variable PLC (of boolean type) that when this one will be at the state " TRUE " will neutralize the button associated to the RECIPE.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the button will become invisible when it will have neutralized.

Button " Color of Bottom " of the group of fields " neutralization "

This button permits to specify the color of bottom of the button of the RECIPE when this one will have neutralized.

Button " Characters " of the group of fields " neutralization "

This button permits to choose the police of character of the text of the button associated to the RECIPE when this one will have neutralized.

Field " Text of the button " of the group of fields " neutralization "

To enter in this field the text to display on the button of the recipe when this one will have neutralized.

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

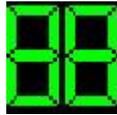
3.2.6.19.2 Billposters 7 segments

This object indicates the value of a variable under a billposter's shape 7 segments.

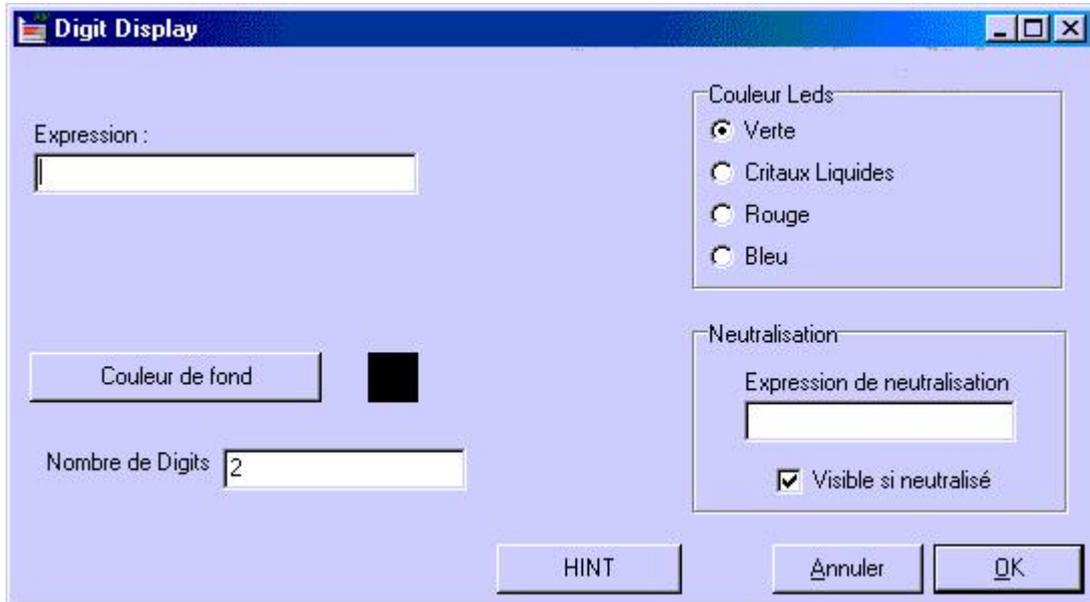
This variable must be the integer or decimal numeric type.

It is possible to define the colors of these segments as well as the color of bottom.

Example of billposter 7 segments :



Dialog box " Billposter 7 segments " :



The parameters are :

Expression

To indicate in this field the name of the variable to visualize with the billposter. This variable must be of integer numeric type (Register).

Example : R12

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !R12

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Color of Bottom

Specify the color of bottom of the billposter. If the color of the leds is " liquid crystals " then this parameter neutralized and took the color gray by default.

Number of Digits

Indicate the number of number that the billposter will have. This parameter must be a integer number.

Example: Variable going from 0 to 99, to put 2 in the number of digits.

Group radio Color Led

This parameter permits to select the color of the leds by checking the box in front of the wanted color. If the box " liquid crystals " is selected the leds then are black on a gray bottom.

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the billposter will become invisible when he will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« Show Hint » Option

The Hint text will be displayed in execution mode only if this option is checked.

« Hint »

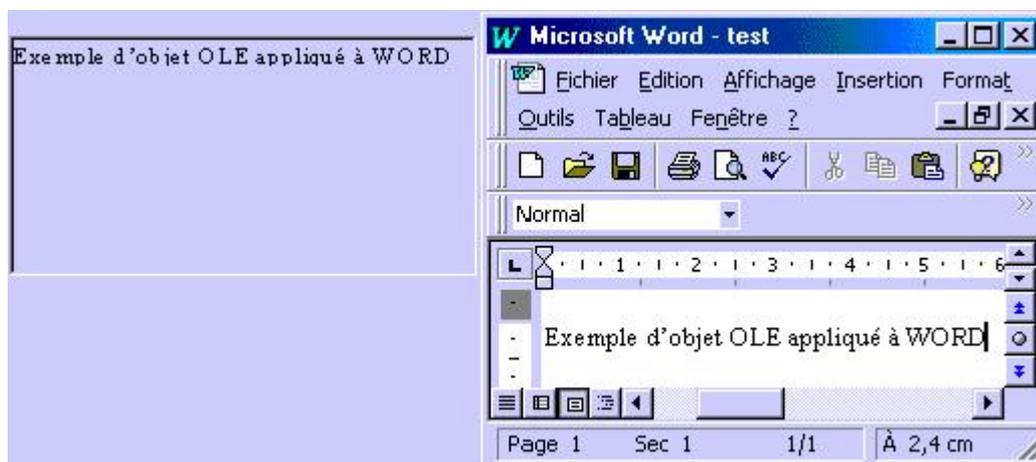
Enter here the "Hint" text. The user can insert a reference to a paramater in the text (example {1}).

3.2.6.19.3 OLE Container

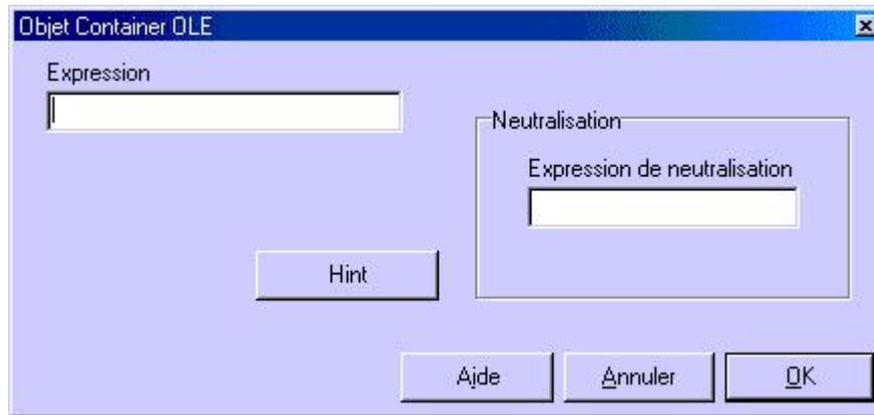
The component OLE Container (OLE container) endows your application with capacities allowing it to do the links and incorporations of objects being on a server OLE.

When you activate an object being in an OLE container, the control passes to the application server OLE. The user can reach all functionalities of the application server then from your application container.

Example of OLE container :



Dialog box "Object OLE Container " :



The parameters are :

Expression

Permits to define the name of the file that it is necessary to insert via an OLE container

To indicate in this field the variable PLC to use (of chained type)

Example : C20

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 !C20

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Expression of neutralization

Permits to specify the variable or the expression of neutralization (This one must be inevitably of boolean type).

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Option " visible if neutralized "

If this option is not checked, the object will become invisible when it will have neutralized.

« Hint » Button

The user can define the "Hint" parameters of the objet.

The object « Hint » text is displayed automatically in a small yellow subwindow when the mouse is moved on the object.

When the user click on the « Hint » button, the following window is displayed.



The parameters are :

« **Show Hint** » **Option**

The Hint text will be displayed in execution mode only if this option is checked.

« **Hint** »

Enter here the "Hint" text. The user can insert a reference to a parameter in the text (example {1}).

3.2.6.19.4 Event Manager

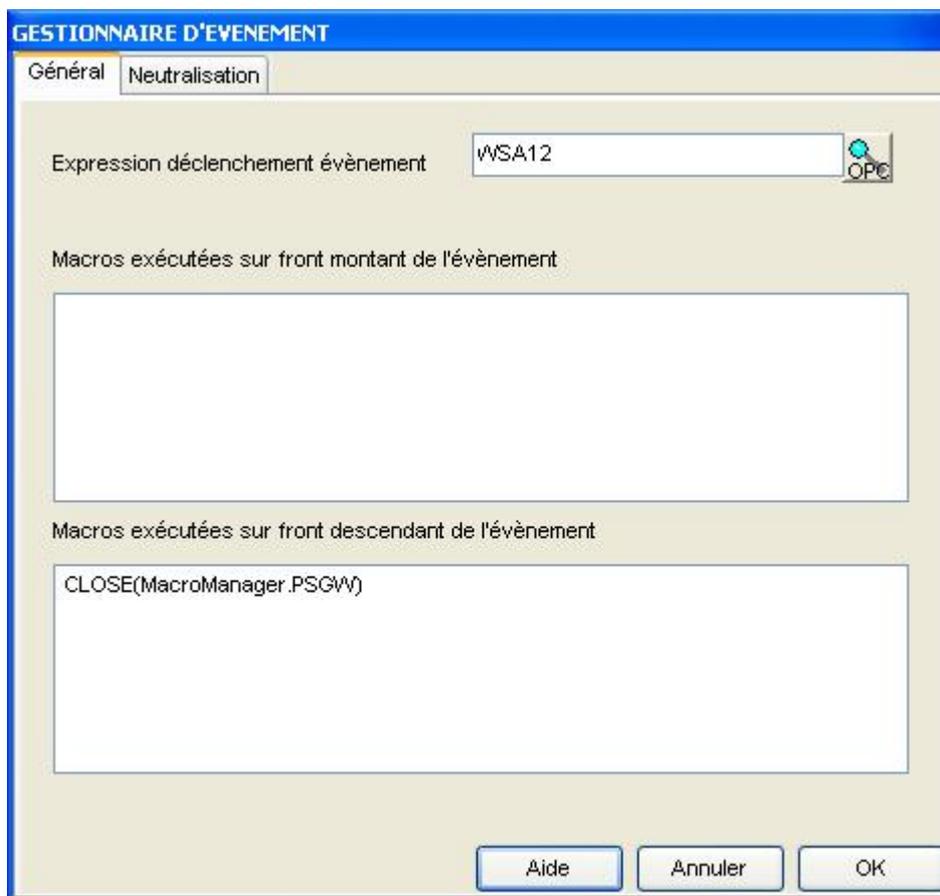
The Event Manager is able to execute macros on an event defined by an expression or a PLC variable of boolean type.

The user can define the macros to execute on a rising edge or on a falling edge of the variable's boolean value.

This object is only visible in design mode



Dialog Box " Event Manager " - General :



The different parameters are :

Expression

Defines the expression or PLC variable that generate the event. The expression type must be boolean.

Example : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Example : !193.0.1.1 ISA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

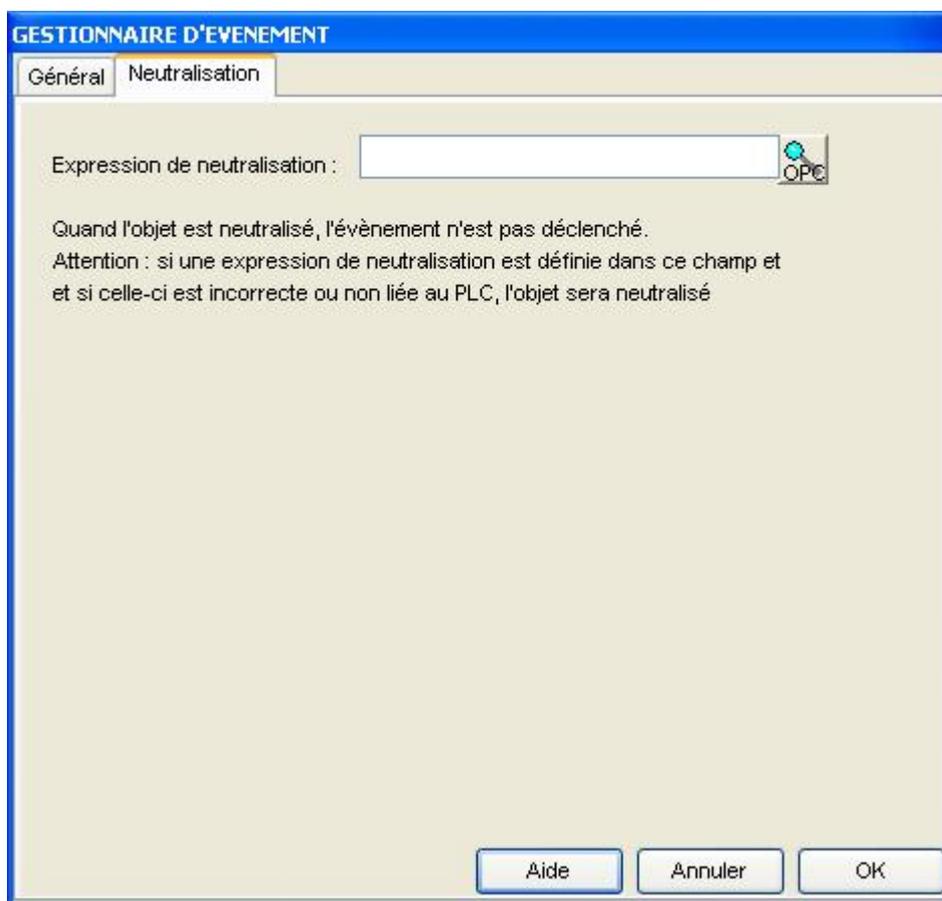
Macros exécutées sur front montant de l'évènement

Entrer ici les différentes macros à exécuter quand la valeur de l'expression passe à 1 (ou True)

Attention : aller à la ligne pour chaque nouvelle macro.

Macros exécutées sur front descendant de l'évènement

Entrer ici les différentes macros à exécuter quand la valeur de l'expression passe à 0 (ou False)

Dialog Box "Event Manager" - Neutralization :

Les paramètres sont :

Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

Exemple : SA10

It's possible to specify the APIGRAF or OPTIMA PLC distant IP in the expression. In this case, the defined IP address that will be taken in account to value the expression.

Exemple : !193.0.1.1 !SA10

It's also possible to insert a reference to a parameter in the field (example: {1} See Section " Open a supervisory form with parameters ")

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

Les évènements ne sont plus déclenchés si la valeur de l'expression de neutralisation est égale à 1 (ou True)

Exemple d'utilisation typique de cet objet

On crée une page TEST.PSGW, on veut que cette page se ferme automatiquement à partir de l'automate. On insère un objet Gestionnaire d'évènement dans la page. On entre la valeur de l'expression SA20 et la macro exécutée sur front montant de l'évènement : CLOSE(TEST.PSGW)

Dès que l'automate OPTIMA PLC va passer SA20 à TRUE, la fenêtre de supervision va se fermer.

3.2.7 Objects explorer

3.2.7.1 Use of the explorer of objects

The explorer of objects is accessible since the slim "Display" -> "Explorer of objects"

The explorer permits to visualize as list hierarchized all present objects on a supervisory form.

After having opened a supervisory form, one can open the explorer of objects :



It is sufficient to click with the right button of the mouse for selection an object in the list. Selected once in the list, the corresponding object of the supervisory form appears selected also.

The explorer of object permits to achieve several operations :

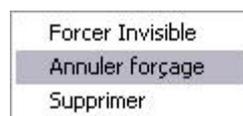
- [To conceal an object in conception](#) ^[285]
- [To conceal all objects in conception](#) ^[285]
- [To annul the masking of an object in conception](#) ^[286]
- [To annul the masking of all objects in conception](#) ^[286]
- [To configure an object](#) ^[286]
- [To suppress an object](#) ^[286]

3.2.7.2 To conceal an object in conception

To open the explorer of object :



To make a right click with the mouse on the object in the list, the menu contextual next one appears :



To choose the menu to "Force invisible." The corresponding object is not more visible on the supervisory form.

3.2.7.3 To conceal all objects in conception

To open the explorer of object :

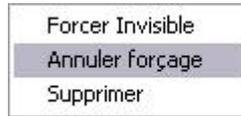


To push on the button "All invisible." All objects of the supervisory form are not more visible.

3.2.7.4 To annul the masking of an object in conception

Open the explorer of object :

To make a right click with the mouse on the object in the list, the menu contextual next one appears :



To choose the menu to "Annul Forçage." The corresponding object becomes again visible on the supervisory form.

3.2.7.5 To annul the masking of all objects in conception

To open the explorer of object :

To push on the button to "Annul Forçages." All objects of the supervisory form that were not more visible become again visible.

3.2.7.6 To configure an object

To open the explorer of object :

To select an object in the list.

To push on the button "to Configure." The window of paramétrage of the object opens up :



3.2.7.7 To suppress an object

To open the explorer of object :

To select the object to suppress in the list.

To push on the button "to Suppress." The object is suppressed then of the supervisory form.

3.2.8 Open a supervisory form from another

The call of a supervisory form from another page is made by using the macro " SUPERG ". (To see the chapter dedicated to the use of the macros).

Example: Call of the page " Page2.psgw "

If the page2.pswg file corresponds then to a supervisory form of the project the call of this page since another can make itself of the following manner :

1. To create a simple button on the main page (page since which one makes the call)
2. To double-click on the button to reach its page of parameterization.
3. In the window " Macro laxity ", to hit the following text: SUPERG (PAGE2)
4. At execution, the forcing of the simple button will have therefore the effect of loading the page " Page2 " to the screen.

Remark: It is possible to call a supervisory form by passing it some parameters. (To see following section " [Call of a supervisory form with passage of parameters](#) ^[287]).

3.2.9 Open a supervisory form with parameters

A supervisory form can be called with parameters. To proceed as follows :

- In the page, to insert some references to parameters with {1}, {2}, etc.
- To call the page by specifying the parameters following the name of file (SUPERG (PAGE, SA10, R20))

Example :

A TEST.PSG page contains two buttons API that must be bound to two boolean variables whose names are only known at the time of the call of the page, for example SA10 and SA11.

In the first button, instead the classic expression, to enter {1}.

In the second button, instead the classic expression, to enter {2}.

In the macro that calls the page, to enter SUPERG (TEST, SA10, SA20) to replace in the page test {1} by SA10 and {2} by SA20.

Other example of application :

The passage of parameters can be used in the case of the user wants to be able to open the same supervisory form while supervising the data of different stations PLC.

In this case, it is necessary in the options of the page, in the field "IP address by default ", to enter {1}.

It is sufficient to call the supervisory form then by passing the IP address to supervise in parameter :

SUPERG (PAGE1,193.0.1.1)

Or SUPERG (PAGE1,193.0.1.2)

3.2.10 Closing a supervisory form with a button

To close the supervisory form in progress, one can use the macro " ENCLOSED ". (To see chapter dedicated to the macros).

1. To create a simple button on the page.
2. To double-click on the button to reach its page of parameterization.
3. In the window " Macro laxity ", to hit the following text: ENCLOSED
4. At execution, the forcing of this button will have the effect of closing the page.

3.2.11 Passwords management

The user has the possibility to use some passwords to permit the opening of some supervisory forms. The password is defined by a number.

To create a to modify a password, one calls on the ChnMDP macro.

Example :

1. In a supervisory form, to create a simple button.
2. In the macro of laxity of the simple button, to hit the macro: ChnMDP (1). To record the page under the name " page1 "
3. In execution makes it to push on this button is going to open a dialog box permitting to define or to modify if this one exists, the password number 1
4. To create a new supervisory form and to record it under the name " page2 "
5. In the properties of the page named " page2 ", to choose the tab OPTIONS. To enter 1 in its input " Password to the opening of the page "
6. In the page1, to create another simple button. To enter the macro of laxity: SUPERG (page2)

7. In execution, to push on the previous simple button. If the password 1 has been defined, for this one will be asked before opening the page 2

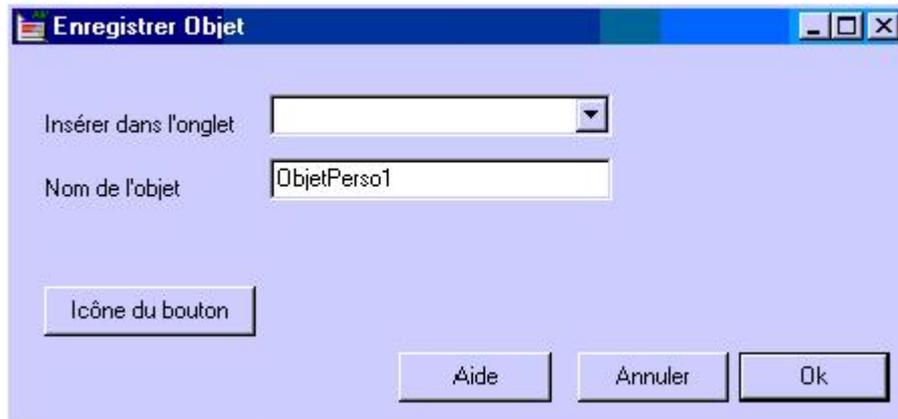
3.3 Objects personalized

3.3.1 Registration of objects Personalized

It is possible to record the personalized objects, that is made some of the classic objects with definite features (color, expression, size.).

A button will be created in an existing tab or no that will permit to recall this object.
To record an object you is necessary to select it and to click in the menu "object " on "to add ", It is also possible to make a right click on the object to make appear the contextual menu and to click on "to record object ".

The menu opens the following window :



To insert in the tab:

This box relates all existing tabs by clicking on the right on the arrow of the box to insert the button of the object to record. It is possible to select a tab in the list or to hit the name of the tab that you created voulé.

Name of the object:

This box permits to bring the name of the button that will be displayed in the Hint in when the cursor will pass above. It defined also the name of the file or will be recorded the object:

(ex: User0ObjetPerso0.bib)

Icon of the button :

The button will have an icon by default:



It is possible to personalize the picture of the icon by clicking on the button and by choosing a file valid picture. A preview of the picture is available in this dialog box type.

3.3.2 Deletion of objects personalized

It is possible to erase the personalized objects. It will erase the button, the created tab if it doesn't have a button more in as well as the file of the object to record.

For it, select an object and made "to suppress " in the menu "object ".
This menu opens the following window :



It is sufficient to select in the lists the objects that you want to suppress and to make 'Ok.'
The paramêtres is they according to:

Object to suppress:

This list relates all objects that have been recorded (objects personalized).

It is sufficient to select the or the objects to suppress. To select several objects, to click on an object and made slip the mouse until the last object that you want to suppress or to click on those that you want while remaining to push on the "Control" key.

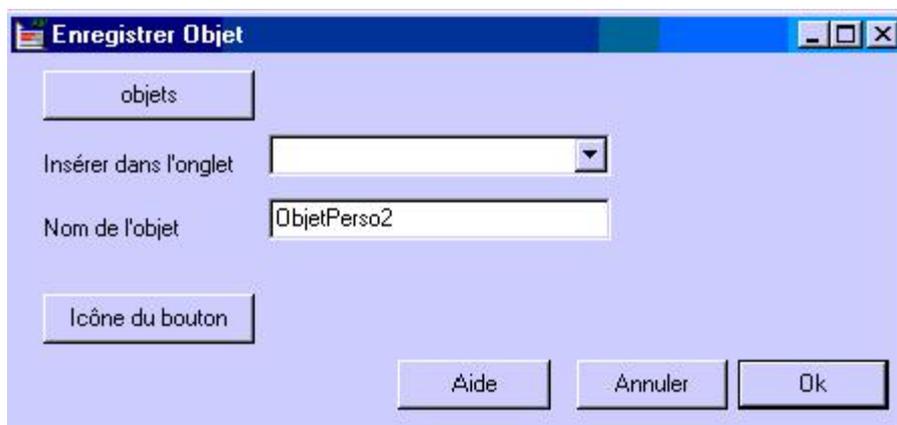
All to select:

Permits to select all objects of the list.

3.3.3 Import of objects personalized

It is possible to import objects personalized thanks to the files of extension " bib ". These files are in " APIGRAFIP\BIN\PERSO\ ".

For it, tops " to import " in the menu " Object ".
This menu opens the following window :



The different paramêtres of this window is:

Objects:

This button permits to choose the corresponding file to the object to import. This file must have the extention "bib."

To insert in the tab:

This box relates all existing tabs by clicking on the right on the arrow of the box to insert the button of the object to record. It is possible to select a tab in the list or to hit the name of the tab that you want to create.

Name of the object:

This box permits to bring the name of the button that will be displayed in the Hint in when the cursor will pass above. It define also the name of the file or will be recorded the object:
(ex: User0ObjetPerso0.bib)

Icon of the button :

The button will have an icon by default:

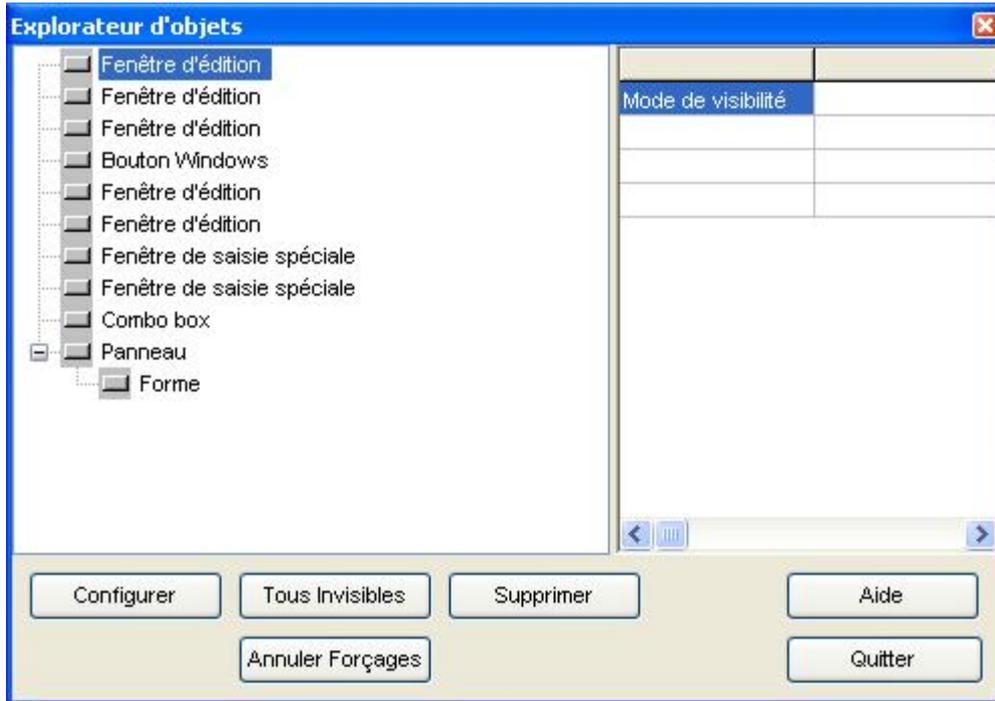


It is possible to personalize the picture of the icon by clicking on the button and by choosing a file valid picture. A preview of the picture is available in this dialog box type.

3.4 Explorer of objects

3.4.1 Explorer of objects

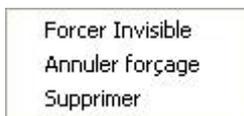
The explorer of objects list in an arborescent window all present objects in the current supervisory form. It is possible since this page to force an object to be invisible during the phase of conception of the supervisory form.



To select an object in the list with the help of the mouse, the corresponding object appears selected then in the supervisory form.

To force an object to be invisible

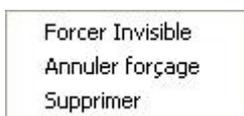
To select an object in the arborescent list and to make a clicks right over then to choose "invisible Focer" in the contextual menu :



The object is not more visible.

To annul to Force an object to be invisible

To select an object in the arborescent list and to make a clicks right over then to choose to "Annul forçage" in the contextual menu :



The object is visible.

- **Button to "Annul Forçages"**

Annul all forçages of invisibility. All objects are then visible in the current supervisory form.

- **Button to "Configure"**

To select an object in the arborescent list and to push on the button "to Configure" to make appear the window of configuration of the object.

- **Button to "Suppress"**

To select an object in the arborescent list and to push on the button "to Suppress" to suppress the selected object.

- **Button "All invisible"**

Force the invisibility of all objects of the current page.

3.5 Use of the macros

3.5.1 Local macros

The local macros are executed on the supervisory workstation (Windows® station)

3.5.1.1 BringToFrontSuperGWindow (local Macro)

Met en avant plan la page de supervision passée en paramètre à condition que la page soit déjà ouverte (avec la macro SUPERG ou éventuellement cachée). Si aucune page n'est passée en paramètre, la page courante est mise en avant plan.

La macro est généralement appelée depuis les macros de relâchement d'un bouton (type bouton de fonction ou autre).

BRINGTOFRONTSUPERGWINDOW

Met en avant plan la fenêtre courante

BRINGTOFRONTSUPERGWINDOW(TEST)

BRINGTOFRONTSUPERGWINDOW(TEST.PSGW)

BRINGTOFRONTSUPERGWINDOW(C:\REP1\TEST)

Affiche la page Test

Remarque : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

BRINGTOFRONTSUPERGWINDOW("C:\Program Files\REP1\TEST")

La macro BRINGTOFRONTSUPERGWINDOW est utilisée généralement conjointement avec les macros HIDESUPERGWINDOW et SHOWSUPERGWINDOW qui permettent d'afficher et de cacher une fenêtre pour permettre à l'utilisateur de gérer les affichages des fenêtres déjà ouvertes.

3.5.1.2 ChnMdp (local Macro)

This macro is used to modify or to create a password in the project.

Each password has a number : 1,2,...

A created password has a new number that can be specified to allow the opening of a supervisory form (See the tab Options of the supervisory form properties window)

Example :

ChnMDP (1)

If the password 1 doesn't exist, the macro will ask for the user to enter the password 1.

If this one exists, the current password will be needed to modify it.

3.5.1.3 ChnUserLanguage (local Macro)

This macro is used to change the current user language in the supervisory (in execution mode). It has only

effects if there several languages defined in the supervisory project.

This macro is generally entered as the releasing macro of a button.

Example :

CHNUSERLANGUAGE (2)

3.5.1.4 Close (local Macro)

Close the current supervisory window. If the supervisory window to close is not the current one, add the supervisory form name to close as a parameter.

Example :

CLOSE

Close the current supervisory form.

CLOSE(TEST)

CLOSE(TEST.PSGW)

CLOSE(C:\REP1\TEST)

CLOSE("C:\Program Files\REPA\TEST")

Close the supervisory form named "TEST". it's not necessary to write the file extension (.PSGW) and to specify the full pathname of the file as the file is in the current project.

Note : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

This macro is typically set as the releasing macro of a function button.

3.5.1.5 CopyPageBitmap (local Macro)

Do a screen copy of the current supervision form into a simple bitmap file.

COPYPAGEBITMAP

The Macro perform a screen copy of the current supervision form into a bitmap file in the supervision directory wich contains the PSGW files. This macro is usually called into the relaxing button macro of any button object.

Example :

With a supervision form named TEST.PSGW, a file named PAGECOPY-TEST-1.BMP will be created at the first call of the macro. At the next call of the macro, a file named PAGECOPY-TEST-2.BMP will be created, and so one.

It's possible to specify a destination directory for the bitmap files.

Examples :

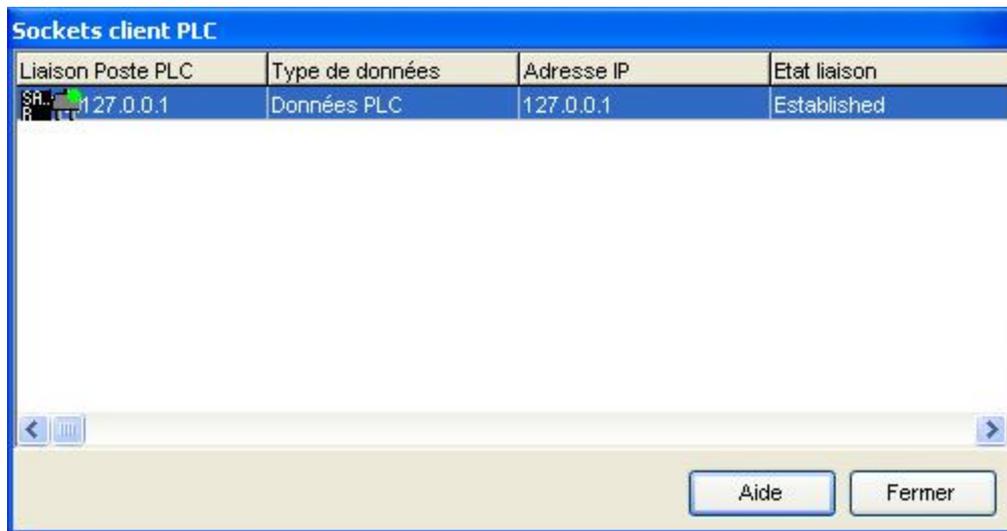
COPYPAGEBITMAP(C:\CURVES) creates BMP files into the directory C:\CURVES

COPYPAGEBITMAP(\CURVES) creates BMP files into the directory C:\...\PSGW\CURVES (Sub driectory CURVES in the supervision directory)

3.5.1.6 DiagSocketPLC

Shows the diagnoses window for the PLC Datas exchange sockets.

The PLC Data sockets are used to value all the variables on OPTIMA PLC or APIGRAF. One socket is created for each OPTIMA PLC/ APIGRAF workstation. If the OPTIMA PLC application is on the same computer, the socket with IP address 127.0.0.1 (Local address) is created.



For more of details, see :
[TCP / IP PLC Datas clients](#)

3.5.1.7 ExecProg (local Macro)

This macro is used to launch an external application (an EXE file)

Example :

EXECPROG (C:\WINDOWS\CALC.EXE)

(See example project in : \EXAMPLES\MACROS)

It's also possible to put some parameters to the command line :

Example :

EXECPROG (C:\WINDOWS\notepad.exe, C:\ESSAI.TXT)

Open the file C:\ESSAI.TXT with the NOTEPAD.EXE Editor.

This macro is typically set as the releasing macro of a function button.

NOTE : Use the double quotes if there are some blanks in the file path.

EXECPROG ("C:\Program files\TEST.EXE")

3.5.1.8 HideCursor (local Macro)

This macro is used to hide the mouse pointer on as supervisory form. It is typically used when the supervisory is on a computer with a touch screen. With this type of computer in some case, it's better to not see the mouse pointer.

To display the mouse pointer, use the "SHOWCURSOR" macro.

This macro is typically set as the opening macro of a supervisory form.

3.5.1.9 HideSuperGWindow (local Macro)

Cache la page de supervision passée en paramètre. Si aucune page n'est passée en paramètre, la page courante est cachée.

La macro est généralement appelée depuis les macros de relâchement d'un bouton (type bouton de fonction ou autre).

HIDESUPERGWINDOW
 Cache la fenetre courante

HIDESUPERGWINDOW(TEST)

HIDESUPERGWINDOW(TEST.PSGW)

HIDESUPERGWINDOW(C:\REP1\TEST)

Remarque : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de

la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

```
HIDESUPERGWINDOW("C:\Program Files\REP1\TEST")
```

3.5.1.10 PrintPage (local Macro)

Thus macro prints the current supervisory form. No parameters are needed.

Example :
PRINTPAGE

This macro is typically set as the releasing macro of a function button.

3.5.1.11 RunHelp (local Macro)

In some supervisory projects, the user make his own help file. (Files with the *.HLP extension). This macro opens the help file in parameter.

Example :
RunHelp (USERHELP.HLP)
This example opens the help file "USERHELP.HLP" in the project current directory. If the help file is not in the project directory, the parameter must contain the full path of the help file .

Example :
RunHelp (C:\HELP\USERHELP.HLP)

This macro is typically set as the releasing macro of a function button.

NOTE : Use the double quotes if there are some blanks in the file path.
RUNHELP ("C:\Program files\USERHELP.HLP")

3.5.1.12 RunHelpIndex (local Macro)

This macro open the index of a user help file with the "*.HLP" extension

Example :
RunHelpIndex (USERHELP.HLP)

If the help file is not in the project directory, the parameter must contain the full path of the help file .

Example :
RunHelpIndex (C:\HELP\USERHELP.HLP)

This macro is typically set as the releasing macro of a function button.

NOTE : Use the double quotes if there are some blanks in the file path.
RUNHELPIINDEX ("C:\Program files\USERHELP.HLP")

3.5.1.13 RunHelpTopic (local Macro)

This macro open a user help file with the "*.HLP" extension at the chapter index specified in the second parameter. (Topic number)

Example :
RunHelpIndex (USERHELP.HLP, 23)
If the help file is not in the project directory, the parameter must contain the full path of the help file .

Example :
RunHelpIndex (C:\HELP\USERHELP.HLP ,23)

This macro is typically set as the releasing macro of a function button.

NOTE : Use the double quotes if there are some blanks in the file path.
RUNHELPTOPIC ("C:\Program files\USERHELP.HLP",32)

3.5.1.14 ShowCursor (local Macro)

This macro shows the mouse pointer. It is only used when the macro "HIDECURSOR" has been executed before and when the mouse pointer is not visible.
It is typically used when the supervisory is on a computer with a touch screen. With this type of computer in some case, it's better to not see the mouse pointer

This macro is typically set as the closing macro of a supervisory form.

3.5.1.15 ShowSuperGWindow (local Macro)

Affiche la page de supervision passée en paramètre à condition que la page soit déjà ouverte (avec la macro SUPERG ou éventuellement cachée). Si aucune page n'est passée en paramètre, la page courante est affichée. La macro est généralement appelée depuis les macros de relâchement d'un bouton (type bouton de fonction ou autre).

SHOWSUPERGWINDOW
Affiche la fenetre courante

SHOWSUPERGWINDOW(TEST)

SHOWSUPERGWINDOW(TEST.PSGW)

SHOWSUPERGWINDOW(C:\REP1\TEST)

Affiche la page Test

Remarque : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

SHOWSUPERGWINDOW("C:\Program Files\REP1\TEST")

La macro SHOWSUPERGWINDOW est utilisée généralement conjointement avec la macro HIDESUPERGWINDOW qui permet de cacher une fenêtre pour permettre à l'utilisateur de gérer les affichages des fenêtres déjà ouvertes.

3.5.1.16 SuperG (local Macro)

Open a supervisory form from another supervisory form.
The only parameter specifies the supervisory file name to open.

SUPERG(TEST)
SUPERG(TEST.PSGW)

If the supervisory form to open is in the current project, the user don't need to specify the extension *.PSGW or the full path of the file.

If the supervisory form is not in the project directory, the parameter must contain the full path of the file :

Example :

SUPERG(C:\TEST\TEST.PSGW)

If there are some blank (space in the file path) the macro must be written like :
SUPERG("C:\Program Files\TEST\TEST.PSGW")

This macro is typically set as the releasing macro of a function button.

3.5.1.17 VisuText (local Macro)

This macro is used to display a text file.

Example :

VisuText (C:\File1.Txt)

NOTE : Use the double quotes if there are some blanks in the file path.
VISUTEXT ("C:\Program files\File1.Txt")

3.5.1.18 WFCopy (local Macro)

This macro is used to copy one or several local files.

The first parameter is the source file(s).

The second parameter is the destination file(s).

Examples :

WFCOPY (C:\ESSAI.TXT, C:\ESSAI.BAK)

WFCOPY (C:\REP1\ * .TXT, C:\REP2\ * .NEW)

NOTE : Use the double quotes if there are some blanks in the file paths.
WFCOPY ("C:\Program files\File1.Txt", "C:\Program files\File1.BAK")

3.5.1.19 WFCreatedDirectory (local Macro)

This macro is used to create a local directory.
 The first parameter is the directory name to create.

Example :

WFCREATEDIRECTORY (C:\REP1)

NOTE : Use the double quotes if there are some blanks in the file paths.
WFCREATEDIRECTORY ("C:\Program files\REP1")

3.5.1.20 WFDelete (local Macro)

This macro is used to delete one or several local files.
 The first parameter is the file name(s) to delete.

Examples :

WFDELETE (C:\REP1\FILE1.TXT)

WFDELETE (C:\REP1\ * .TXT)

NOTE : Use the double quotes if there are some blanks in the file paths.
WFDELETE ("C:\Program files\REP1\FILE1.TXT")

3.5.1.21 WFDeleteDirectory (local Macro)

This macro is used to remove a local directory and its content.
 The first parameter is the directory name to delete.

Example :

WFDELETEDIRECTORY (C:\REP1)

NOTE : Use the double quotes if there are some blanks in the file paths.
WFDELETEDIRECTORY ("C:\Program files\REP1")

3.5.1.22 WFRename (local Macro)

This macro is used to rename a local file or a local directory.
 The first parameter is the name of the file or the directory to rename.
 The second parameter is the new name of the file or the directory.

Examples :

WFRENAME (C:\REP1\FILE1.TXT, C:\REP1\NEWFILE1.TXT)

WFRENAME (C:\OLDREP, C:\NEWREP)

NOTE : Use the double quotes if there are some blanks in the file paths.
WFRENAME ("C:\Program files\OLDREP1", "C:\Program files\OLDREP2")

3.5.1.23 WFSetFileAttr (local Macro)

This macro is used to modify the attributes of a local file.
 The first parameter is the file to which to apply new attributes.
 The second parameter is the attribute to apply to the file. This parameter is an integer value.

Examples :

WFSETFILEATTR (C:\REP1\FILE1.TXT ,36)

This macro transforms the File1.txt file in file system (hidden file).

WFSETFILEATTR (C:\REP1\FILE1.TXT ,32)

This macro returns the normal attributes to the File1.Txt file (Attributes: normal)

NOTE : Use the double quotes if there are some blanks in the file paths.
WFSETFILEATTR ("C:\Program files\OLDREP1", 32)

3.5.1.24 WNetSChangeFileName (local Macro)

This macro is used to rename a distant file on a APIGRAF WORKSTATION or an OPTIMA PLC workstation running the APIFTPSEVER service
 The first parameter (optional) is PLC workstation IP address.
 The second parameter is the file name.
 The third parameter is the new name of the file.

Examples :

WNETSCCHANGEFILENAME (193.0.1.1, C:\OLD.TXT, C:\NEW.TXT)

WNETSCCHANGEFILENAME (C:\OLD.TXT, C:\NEW.TXT)

The second example uses the default IP address of the project because no IP Address is specified in the parameters

3.5.1.25 WNetSCopyFileRemote (local Macro)

This macro permits to copy a file of a station PLC toward another station PLC.

The first parameter designates the IP address of the station containing PLC the file source.

The second parameter indicates the name of the file source.

The third parameter indicates the IP address of the destination station PLC.

The fourth parameter designates the name of the destination file.

Example :

WNETSCOPYFILEREMOTE (193.0.1.1, C:\OLD.TXT ,193.0.1.2, C:\NEW.TXT)

3.5.1.26 WNetSCreateDirectory (local Macro)

This macro permits to create a directory on a station PLC.

The first parameter is optional, it designates the IP address of the station PLC.

The second parameter indicates the name of the directory to create.

Examples :

WNETSCREATEDIRECTORY (193.0.1.1, C:\REP1)

WNETSCREATEDIRECTORY (C:\REP1)

The second example uses the address by default distant IP of the project.

3.5.1.27 WNetSDeleteDirectory (local Macro)

This macro permits to suppress a directory on a station PLC.

The first parameter is optional, it designates the IP address of the station PLC.

The second parameter indicates the name of the directory to suppress.

Examples :

WNETSDELETEDIRECTORY (193.0.1.1, C:\REP1)

WNETSDELETEDIRECTORY (C:\REP1)

The second example uses the address by default distant IP of the project.

3.5.1.28 WNetSReadAndMoveFile (local Macro)

This macro permits to copy a file of a station PLC toward the WIN station that calls this macro. To the continuation of the copy, the file source is suppressed on the station PLC.

The first parameter is optional, it designates the IP address of the station containing PLC the file source.

The second parameter indicates the name of the file source on the station PLC.

The third parameter indicates the name of the destination file on the WIN station.

Examples :

WNETSREADANDMOVEFILE (193.0.1.1, C:\FILE1.TXT, C:\REP1\FILEDOS1.TXT)

WNETSREADANDMOVEFILE (C:\FILE1.TXT, C:\REP1\FILEDOS1.TXT)

The second example uses the address by default distant IP of the project.

3.5.1.29 WNetSWriteAndMoveFile (local Macro)

This macro permits to copy a file of the WIN station that calls this macro toward a station PLC. To the continuation of the copy, the file source is suppressed on the WIN station.

The first parameter is optional, it designates the IP address of the station PLC on which one must do the copy.

The second parameter indicates the name of the file source on the WIN station.

The third parameter indicates the name of the destination file on the station PLC.

Examples :

WNETSWRITEANDMOVEFILE (193.0.1.1, C:\FILE1.TXT, C:\REP1\FILEWIN1.TXT)

WNETSWRITEANDMOVEFILE (C:\FILE1.TXT, C:\REP1\FILEWIN1.TXT)

The second example uses the address by default distant IP of the project.

3.5.1.30 WNetSReadFile (local Macro)

This macro permits to copy a file of a station PLC toward the WIN station that calls this macro.

The first parameter is optional, it designates the IP address of the station containing PLC the file source.

The second parameter indicates the name of the file source on the station PLC.

The third parameter indicates the name of the destination file on the WIN station.

Examples :

WNETSREADFILE (193.0.1.1, C:\FILE1.TXT, C:\REP1\FILEDOS1.TXT)

WNETSREADFILE (C:\FILE1.TXT, C:\REP1\FILEDOS1.TXT)

The second example uses the address by default distant IP of the project.

3.5.1.31 WNetSWriteFile (local Macro)

This macro permits to copy a file of the WIN station that calls this macro toward a station PLC.

The first parameter is optional, it designates the IP address of the station PLC on which one must do the copy.

The second parameter indicates the name of the file source on the WIN station.

The third parameter indicates the name of the destination file on the DOS station

Examples :

WNETSWRITEFILE (193.0.1.1, C:\FILE1.TXT, C:\REP1\FILEWIN1.TXT)

WNETSWRITEFILE (C:\FILE1.TXT, C:\REP1\FILEWIN1.TXT)

The second example uses the address by default distant IP of the project.

3.5.1.32 WNetSSetFileAttr (local Macro)

This macro permits to modify the attributes of a file on a station PLC.

The first parameter is optional, it designates the IP address of the station PLC on which one must do the copy.

The second parameter indicates the name of the file on the station PLC.

The third parameter the attribute to apply to the file of the station PLC. This parameter is of integer type.

Examples :

WNETSSETFILEATTR (193.0.1.1, C:\FILE1.TXT ,32)

WNETSSETFILEATTR (C:\FILE1.TXT ,32)

The second example uses the address by default distant IP of the project.

3.5.2 External or distant macros (executed on the PLC)

The external macros are executed on the station kernel real time, these macros are the macros standard of DOS APIGRAF.

The DOS macros " exploitable " since the Windows® supervisory are described in the following chapters.

The external macros are executed on the kernel APIGRAF PLC whose IP address corresponds either to the IP address of the supervisory form of where is done the macro (If an address to been specified for the page) is on the APIGRAF kernel whose IP address corresponds by default to the IP address of the project.

It is however possible to specify l 'IP address of the kernel APIGRAF PLC directly in the text of the local macro.

Example :

SETSA1=1 (Set of the variable SA1 on the kernel APIGRAF PLC of the IP address by default of the page or the project).

!193.0.1.1!SETSA1=1 (Set of the variable SA1 on the kernel APIGRAF PLC whose IP address is 193.0.1.1).

3.5.2.1 BEEP (Macro PLC)

Provoke a beep on the loudspeaker of the PC.

3.5.2.2 DECR ###(Macro PLC)

Decrement of the register R No " ### "

Example :

DECR10

3.5.2.3 INCR ###(Macro PLC)

Increment of the register R No " ### "

Example :

INCR10

3.5.2.4 INVSA ###(Macro PLC)

Inversion of the state of the SA No " ### "

Example :

INVSA5

3.5.2.5 SET...= (Macro PLC)

Affectation of a value to a variable PLC.

Examples :

```
R15=100 SET
R15=R10+R11 SET
L10=1 SET
SA23=0 SET
```

3.5.3 User macros

3.5.3.1 Generalities

APIGRAF WIN offers the possibility to the users to develop their own macros. These macros will be able to be executed then since the supervisory.

The macros users are developed in DLL " user " who are create with the help of DELPHI®.

3.5.3.2 Development of a User DLL

Afin de développer une DLL utilisateur, on doit disposer d'un compilateur DELPHI 32 bits (ou autre)

A l'aide de delphi, créer un projet de type DLL.

La première unité mentionnée dans le USES de la DLL doit absolument être SHAREMEM

Dans le menu « Outils -> Options du débogueur », choisir l'onglet « Exceptions du langage » et décocher la case « Arrêter sur exceptions Delphi » si la case correspondante est cochée.

L'architecture de la DLL doit respecter le format suivant :

Architecture d'une DLL Utilisateur :

```
LIBRARY LIBUSER ;
USES SHARMEM ;
{$R *.RES}
type
  TParamsList = class(TStringList)
    private
      vMacro : string;
    public
      EndMacro : string;
      property Macro : string read vMacro;
    end;
Function GetMacroNumber : Integer ; Stdcall ;
begin
  ...
end ;
Function GetMacroName (AmacroNum : Integer ;
  Var AMacroName : string ;
  Var AmacroAddress : pointer)
  : Boolean ; stdcall ;
begin
  ...
end ;
function GetItemInfo ( Anum : Integer;
  var AType : Integer;
  var AStrID : shortstring;
  var AIPAddress : shortstring;
  var AOPCServerName : shortstring;
  var AAddress : pointer )
  : boolean; stdcall; export;
begin
  ...
end ;
exports
  GetMacroNumber ,
  GetMacroName ,
```

```
GetItemInfo;
```

End.

La Fonction **GetMacroNumber** doit renvoyer le nombre de macros.

La Fonction **GetMacroName** doit renvoyer en paramètres pour un numéro de macro donnée (AmacroNum allant de 1 au nombre de macros présentes dans la DLL ; le nombre de macros présentes dans la DLL est retourné par la fonction GetMacroNumber) le nom de la macro dans AmacroName et l'adresse de procédure correspondant à la macro dans AmacroAddress sous forme de pointeur.

La fonction **GetItemInfo** doit renvoyer en paramètres les infos concernant les variables de la DLL à mettre à jour avec les variables automates

Les paramètres sont les suivants :

AType : spécifie le type de variable à relier à la variable automate

- 1 : type boolean
- 2 : type Longint
- 3 : type real
- 4 : type Extended
- 5 : type shortstring

AStrID : désignation de la variable automate à superviser

Exemple : 'R10'

AIPAddress : Adresse IP du poste PLC sue lequel on doit superviser la variable.

Exemple : '193.0.1.1'

AAddress : pointeur sur la variable à relier à la variable automate

Exemple : @long1

Exemple de fonction GetItemInfo :

Dans cet exemple on veut utiliser deux variables DELPHI (Long1 de type longint et Bool1 de type Booléen). On veut que la valeur de Long1 soit mise à jour avec la valeur de la variable automate R10 à l'adresse IP 193.0.1.1 De même, on désire que la valeur deBool1 soit mise à jour avec la valeur de la variable automate SA10 à l'adresse IP 193.0.1.1

Une modification de la valeur de Long1 ou de la valeur de Bool1 entraînera une mise à jour de la valeur de R10 ou de SA10.

La fonction GetItemInfo doit retourner TRUE pour Anum=1 et Anum=2 et FALSE sinon, puisqu'il y a 2 variables à superviser (R10 et SA10).

Attention : il faut bien respecter les types des variables

```
function GetItemInfo ( Anum           : Integer;
                      var AType       : Integer;
                      var AStrID      : shortstring;
                      var AIPAddress  : shortstring;
                      var AOPCServerName : shortstring;
                      var AAddress    : pointer   )
: boolean; stdcall; export;

begin
Result := True ;
AOPCServerName := '';
Case Anum of
  1      : begin
            AIPAddress := '193.0.1.1' ;
            Atype      := 2 ;
            AStrID     := 'R10' ;
            Aaddress   := @Long1 ;
            end ;
  2      : begin
            AIPAddress := '193.0.1.1' ;
            Atype      := 1 ;
            AStrID     := 'SA10' ;
            Aaddress   := @Bool1 ;
            end ;
end ;
```

```

    else begin
        Result := False;
    end ;
End ;
end ;

```

Format d'une macro Utilisateur

```

Procédure MacroUser1 ( Params : TparamsList ) ;
    Begin
        ...
    end ;

```

L'unique paramètre des macro doit être de type TparamsList (C'est une liste de chaines)

Le nombre de paramètres est ici donné par Params.Count

L'accès aux paramètres (qui sont se type string) est réalisé avec :

Params[0],...,Params[Params.Count - 1]

Exemple d'appel d'une macro utilisateur depuis une page de supervision.

```
MacroUser1(Chaine1,12)
```

Dans cet exemple on passe 2 paramètres sous forme de string, la macro utilisateur doit se charger de la conversion des types.

In order to develop a DLL user, one must have a DELPHI compiler 32 bits (or other)

With the help of delphi, to create a project of DLL type.

The first unit mentioned in USE it of the DLL must be absolutely SHAREMEM

In the menu " Tools -> Options of the debugger ", to choose the tab " Exceptions of the language " and to uncheck the box " To stop on Delphi exceptions " if the corresponding box is checked.

The architecture of the DLL must respect the following format :

Architecture of a DLL User :

```

LIBRARY LIBUSER ;
USES SHARMEM ;
{$R *.RES}
type
    TParamsList = class(TStringList)
        private
            vMacro : string;
        public
            EndMacro      : string;
            property Macro : string read vMacro;
        end;
Function GetMacroNumber : Integer ; Stdcall ;
    begin
        ...
    end ;
Function GetMacroName (AmacroNum : Integer ;
                        Var AMacroName : string ;
                        Var AmacroAddress : pointer)
                        : Boolean ; stdcall ;
    begin
        ...
    end ;
function GetItemInfo ( Anum          : Integer;
                        var AType      : Integer;
                        var AStrID     : shortstring;
                        var AIPAddress : shortstring;
                        var AOPCServerName : shortstring;
                        var AAddress   : pointer )
                        : boolean; stdcall; export;
    begin

```

```

...
end ;
exports
  GetMacroNumber ,
  GetMacroName ,
  GetItemInfo ;

```

End.

The Function **GetMacroNumber** must send back the number of macros.

The Function **GetMacroName** must send back in parameters for a number of macro data (active AmacroNum of 1 to the number of macros presents in the DLL; the number of macros presents in the DLL returned by the GetMacroNumber function) the name of the macro in AmacroName and the address of procedure corresponding to the macro in AmacroAddress as drawer.

The function **GetItemInfo** must send back in parameters the infos concerning the variables of the DLL to put up to date with the variable PLCs

The parameters are the next one :

AType : specify the type of variable to join to the variable PLC

- 1: type boolean
- 2: Longint type
- 3: type real
- 4: Extended type
- 5: type shortstring

Astrid : designation of the variable PLC to supervise

Example: 'R10'

AIPAddress : IP address of the station PLC sweets which one must supervise the variable.

Example: '193.0.1.1'

AAddress : drawer on the variable to join to the variable PLC

Example: @ long1

Example of GetItemInfo function :

In this example one wants to use two variable DELPHI (Long1 of type longint and Bool1 of boolean type). One wants that the value of Long1 is put up to date with the value of the variable R10 PLC to the IP address 193.0.1.1 In the same way, one wants that the value deBool1 is put up to date with the value of the variable SA10 PLC to the IP address 193.0.1.1

A modification of the value of Long1 or the value of Bool1 will entail an update of the value of R10 or SA10.

The GetItemInfo function must return TRUE for Anum=1 and Anum=2 and FALSE otherwise, since there are 2 variables to supervise (R10 and SA10).

Attention: it is necessary to respect the types of the variables

```

function GetItemInfo ( Anum           : Integer;
                      var AType       : Integer;
                      var AStrID      : shortstring;
                      var AIPAddress  : shortstring;
                      var AOPCServerName : shortstring;
                      var AAddress    : pointer      )
                      : boolean; stdcall; export;

begin
  Result := True ;
  AOPCServerName := '';
  Case Anum of
    1      : begin
              AIPAddress := '193.0.1.1' ;
              Atype      := 2 ;
              AstrID     := 'R10' ;
              Aaddress   := @Long1 ;
            end ;
    2      : begin
              AIPAddress := '193.0.1.1' ;
              Atype      := 1 ;
            end ;
  end ;
end ;

```

```

        AstrID := 'SA10' ;
        Address := @Booll ;
        end ;
    else begin
        Result := False;
        end ;
End ;
end ;

```

Format of one user macro

```

Procedure MacroUser1 ( Params : TparamsList ) ;
    Begin
        ...
    end ;

```

The unique parameter of the macro must be of TparamsList type (It is a list of chains)

The number of parameters is given here by Params.Count

The access to the parameters (which are strings) is achieved with :

Params [0].., Params [Params.Count . 1]

Example of one user macro call since a supervisory form.

```
MacroUser1 (Chaine1,12)
```

In this example one passes 2 parameters as string, the user macro must be in charge of the conversion of the types.

3.5.3.3 User DLL example

The source of the DLL as well as the DLL are available in the DLLUSER directory of the directory Examples.

The following example illustrates a DLL user having 3 macros user: power, messageinfo and sum.

```

library LibUser;
uses sharemem,
    classes,
    SysUtils,
    dialogs,
    windows;
{$R *.RES}
type
    TparamsList = class(TStringList)
        private
            vMacro      : string;
        public
            EndMacro    : string;
            property Macro : string read vMacro;
        end;

var
    X1  : Longint;
    X2  : Longint;
    X3  : Longint;
    X4  : Longint;
    X5  : Longint;
    X6  : Longint;
    X7  : Longint;

const
    NbMacro = 3;

function GetItemInfo ( Anum      : Integer;
                        var AType  : Integer;
                        var AStrID : shortstring;
                        var AIPAddress : shortstring;
                        var AAddress : pointer )
    : boolean; stdcall; export;

begin
    Result := True;
    AIPAddress := '193.0.1.1';
    Case Anum of
        1 : begin

```

```

        AAddress := @X1;
        AType    := 2;
        AStrID   := 'R12';
    end;
2 : begin
    AAddress := @X2;
    AType    := 2;
    AStrID   := 'R13';
    end;
3 : begin
    AAddress := @X3;
    AType    := 2;
    AStrID   := 'R14';
    end;
4 : begin
    AAddress := @X4;
    AType    := 2;
    AStrID   := 'R15';
    end;
5 : begin
    AAddress := @X5;
    AType    := 2;
    AStrID   := 'R16';
    end;
6 : begin
    AAddress := @X6;
    AType    := 2;
    AStrID   := 'R20';
    end;
7 : begin
    AAddress := @X7;
    AType    := 2;
    AStrID   := 'R21';
    end;
else begin
    Result := False;
    Exit;
end;
end;{Case}
end;
Procedure Somme(Params : TParamsList);
begin
    X1 := X2 + X3 + X4 + X5;
end;
Procedure MessageInfo(Params : TParamsList);
begin
    if Params.Count <= 0
    then Exit;
    MessageDlg(Params[0],mtinformation,[MbOK],0);
end;
Procedure Puissance(Params : TParamsList);
var    Pui    : Integer;
       Res    : Integer;
       i      : Integer;
begin
    if Params.Count <= 0
    then Exit;
    try
        pui := StrToInt(Params[0]);
    except
        Exit;
    end;
    if Pui <= 1
    then Exit;
    Res := 1;
    for i:= 1 to pui
    do begin
        Res := Res * X7;
        end;
    X6 := Res;

```

```

    end;
{Fonction qui retourne le nombre de macros présentes}
Function GetMacroNumber : Integer; stdcall; export;
    begin
        Result := NbMacro;
    end;
Function GetMacroName (AMacroNum : Integer;
                        var AMacroName : String;
                        var AMacroAddress : Pointer) :
                        Boolean; stdcall;

    begin
        Result := False;
        if (AMacroNum < 1)
            or (AMacroNum > NbMacro)
            then Exit;
        Case AMacroNum of
            1 : begin
                AMacroName      := 'Somme';
                AMacroAddress    := @Somme;
            end;
            2 : begin
                AMacroName      := 'MessageInfo';
                AMacroAddress    := @MessageInfo;
            end;
            3 : begin
                AMacroName      := 'Puissance';
                AMacroAddress    := @Puissance;
            end;
        end;{Case}
        Result := True;
    end;
{ La DLL doit obligatoirement exporter ces deux fonctions }
{ afin d'être utilisé dans APIGRAF WIN }
exports
    GetMacroNumber,
    GetMacroName,
    GetItemInfo;
end.

```

3.5.3.4 Using a user DLL in a project

Once the dll to been compiled, it is necessary to specify to the project that it must use the DLL.

In the menu Macros -> Macros User -> to Ask DLL User, it is necessary to specify the DLL to charge. If the DLL to been charged correctly and that this one is formatted correctly, the available macros in this DLL appear henceforth in the lists local macros (to SEE Menu Macros -> local Macros) and are accessible in the project. The associated commentary has these macros in the list makes appear the name of DLL file and the number of macro of the file.

Once loaded, the dll will automatically be reloaded to every opening of the project or has every execution of a supervisory form of the project. The name of the loaded DLL is consigned indeed in the file *.INI of the project.

3.5.3.5 Unload a user DLL in a project

To suppress a DLL user of a project, it is necessary to go in the menu macros -> Macros User -> to Unload a DLL User.

The list of the DLL User appears then.

To choose in the list the dll to suppress and to click on the button " To unload ".

The dll won't be then more loaded in the project.

3.5.3.6 User DLL list

The menu Macros -> Macros User -> List of the DLL User permits to visualize the list of all DLL user charged in the project. For every DLL, the number of macros is indicated.

To visualize the available macros in every DLL, it is necessary to refer to the menu Macros -> local Macros that permit to visualize all available local macros. The commentary associated to the macros specifies if these are charged in DLL User.

3.6 Translations & Languages management

3.6.1 Language files

It is possible to have some supervisory forms in several languages. For it is necessary before all to create files of language. These files of language will contain the different translations.

The names of the language files are the LANGAGE.1 type, LANGAGE.2, etc. Chaque file language will be affected to a language. The files should be present in the directory of the project.

3.6.1.1 Creating a new language file

To choose in the menu " Tools ", " Translations user " then " New file of language ".

So no file language was present, the name of the file created is " LANGAGE.1 "

The translations should be enter the following way in the file :

```
@ LG1 :
Example of translation
@ LG2 :
N°2 text
@ LG3 :
N°3 text
.. Etc
```

In this example, if one wants the correspondence in English, it will be necessary to create a new file language " LANGAGE.2 " :

```
@ LG1 :
Transfer example
@ LG2 :
Text #2
@ LG3 :
Text #3
```

3.6.1.2 Using language files in the supervisory forms

If one wants for example that the text of a button is translated, it is necessary to enter in the parameter text of the button, a code of the Type @ LG2.

You can also associate a commentary to the code (example: @ LG2:Bouton Page 2). In this case the text associated to the button will be listed in the file language to the following level :

```
@ LG2 :
N°2 text
```

3.6.1.3 Selecting a language

Once the files complete languages and the supervisory forms coded with the symbol @ LG, the CHNUSERLANGUAGE macro by passing it in parameter the number of the language file to use.

Example: CHNUSERLANGUAGE (2)
This macro can be placed for example in a button.

3.6.1.4 Translation example

A project of example is available in..\APIGRAFIP\EXEMPLES\TRADUCTION.

This project contains two files of LANGAGE.1 language and LANGAGE.2.

3.7 Users Management

3.7.1 Generalities

It is possible to restrict the possibilities of action in the runtime according to the user. For it, the mode users management must be activated in the menu Options / Users management.

In this mode, for a user name and a password can be asked before giving access to a supervisory form, according to the group of adherence of the user.

In this case, it is necessary to define in the properties of the supervisory form a minimum level for the access to this one.

The groups of users are the next one :

--	--

Groups	Rights
Admin	Complete rights.
Maintenance	Complete rights, except for the users management and the modification of the working options.
Supervisors	Access limited to the creation of operators, to the modification of the variable standards of the PLC (R, SA, etc.) and to the visualization of the set of the information.
Operators	Access limited to the visualization of some information (variable standards and inputs/outputs).

By default, the users management is not activated, and the access is equivalent to the Admin group. In case of activation of the users management, if this activation is made for the first time, only the "Admin" user, of the Admin group, is predefined. The initial password is "Admin."

In case of loss of the passwords preventing the access to the runtime, the deletion of the Users.dat file situated in \Bin (directories of installation of optima View) permits to take the described above initial configuration.

3.8 Execution of the supervisory forms

3.8.1 Files create in a project of supervisory

At the time of the creation of a new project, a file " NomProjet.INI " is created in the project. This file contains some information concerning the project.

A copy of this file named "NomProjet.INB " can be also present in the directory of the project. So no of these files is not present in the directory, the information contained in these will be lost and an error message warns the user. A new file * .INI will be created then to the closing of the page.

3.8.2 Execution of one page since the workshop

To execute a supervisory form since the publisher, it is necessary to open the page that one wishes to execute in the project first of all.

Then, it is necessary to click on the icon " Execution " of the toolbar floating " FUNCTIONS ". This function is symbolized by a gearing.

3.8.3 Execution of one page with its icon

A supervisory form is protected in a file * .PSWG. This type of file extension is bound to the executable of the SUPERGW.EXE supervisory forms.

To execute a page directly, it is sufficient to double-click on its icon therefore.

3.9 Virtual keyboard

3.9.1 Use of the virtual keyboard

The virtual keyboard is used on the tactile screens that are not fit with keyboard. It is a keyboard that opens up to the screen to quoted of the input window and that permits to the user to enter alphanumeric characters by clicking on the buttons of this keyboard.

This option permits to open the virtual keyboard of automatic manner of the that the user clicks on a input window in a supervisory form.

Example : Alphanumeric virtual keyboard



To put the virtual keyboard in place to see the menu :

[Menu general Options](#) ^[89]

3.10 Use of DDE ties

3.10.1 General principles

The launching of a supervisory form out of the supervisory workshop or the launching of the application " Supervisory workshop " provoke the launching of a server of data (APIDATAS.EXE). This server of data can be used by server DDE APIGRAF IP: APIDDE.EXE. All data PLCs can be exploited therefore in Windows® applications as Excel, Word...

Server DDE is characterized by :

- The containing application the server: APIDDE
- The server's name or (Topic): PLCDATA
- The server's categories (or Item): These are created at the time of the demand of tie on a variable PLC particular on a kernel APIGRAF PLC of IP address data

Example : !193.0.1.1 !R12

This item permits to recover the value of the R12 register on the kernel PLC of IP address 193.0.1.1.

3.10.2 Example: Creation of a DDE tie with MICROSOFT® Excel

In a new Excel document, hit in a cell the following formula :

=APIPDE|PLCDATA!'!193.0.1.1 !R12'

Validate then with " ENTERED "

If server APIDDE is not thrown, EXCEL® will propose the casting.

The cell displays the value of R12 then. (If server APIDDE.EXE is thrown well)

An example illustrating the DDE ties with Excel® is in the subdirectory \EXAMPLES\Lien DDE Excel. In the directory of this example one file is containing Excel a diagram to illustrate the DDE ties.

3.10.3 Authorization of modification of variables PLCs via a DDE tie

To allow the modification of a variable PLC via a DDE tie, it is necessary to specify it in the writing of the category :

Example :

Category not allowing the modification: !193.0.1.1 !R12

Category allowing the modification: M!193.0.1.1 !R12

In the example of the previous section, it would result in the formula :

=APIPDE|PLCDATA!'M!193.0.1.1 !R12'

3.11 Access to the variable PLCs since a program writes in DELPHI

3.11.1 General principles

At the time of the installation of IP APIGRAF, two files concerning a packet of execution DELPHI 5 are installed.

The USERAPIPLC.BPL file is copied in the C:\WINDOWS\SYSTEM directory

The USERAPIPLC.DCP file is copied in the C:\PROGRAM directory FILES\OPTIMALOG\APIGRAFIP\BIN

\Paquet-Delphi 5

A packet of DELPHI execution permits to reach functions as declaring the main unit of the packet merely in the clause USES the program. In the present case, it is sufficient to make: USE UClientAPI ;

The packet of execution being compiled, it is solely usable with DELPHI5.

3.11.2 Creation of a DELPHI5 program using the USERAPIPLC packet

To create a DELPHI5 program using the USERAPIPLC.DCP packet, it is necessary to follow the following stages :

1. Créer a new application in DELPHI5 and to copy the USERAPIPLC.DCP file in the directory of your project DELPHI.
2. Dans the menu of DELPHI: Tools - > Options of the debugger, to choose the tab " Exceptions of the language ". In that tab, to uncheck the option " To stop on DELPHI exceptions " if this one is checked.
3. Dans the menu of DELPHI: Project - > Options, to choose the tab " Packets ". In this tab, to check the option " to construct with the packets of execution ". In the group " Packets of execution ", to click on the button to Add then on to browse to choose the USERAPIPLC.DCP packet that has been copied in the directory of your DELPHI project.
4. Dans the clause USES your main page, to insert the name of the UClientAPI unit.
5. A this stage, you can use the functions of the packet described following section.

3.11.3 Accessible functions since the USERAPIPLC packet

3.11.3.1 LinkToAPI function

Function LinkToAPI (AAddress: to point; AIDStr: shortstring; AIPAddress: shortstring; AdataType: Integer): Boolean ;

This function permits to join a variable DELPHI to a variable PLC. Every time that the variable PLC changes state, the variable DELPHI is put up to date with the value of the variable PLC. Inversely, a modification of the value of the variable DELPHI will entail an update of variable PLC with the value of the variable DELPHI.

Parameters :

Aaddress : drawer on the variable DELPHI to cool with a variable PLC

AIDStr : Chain specifying the variable PLC to join

AIPAddress : IP address of the station PLC

AdataType : type of the variable DELPHI. The possible choices are :

- | | |
|---|---------------|
| 1 | : Boolean |
| 2 | : Longint |
| 3 | : Real |
| 4 | : Extended |
| 5 | : ShortString |

Result of the function :

The function sends back TRUE if the demand has been taken well in account and FALSE otherwise.

Attention: it doesn't mean that if the function sends back TRUE, then the variable DELPHI is cooled well with the value of the variable PLC. It is necessary to use the IsLinkedToAPI function or AllLinksOK for the knowledge.

Example :

```
LinkToAPI (@ X1, 'R466', '193.0.1.1', 2) ;
```

(With X1: Longint)

Remark : If the variable is of local type, the parameter IP addresses can be empty :

```
LinkToAPI (@ X1, 'WR1', "");
```

To see [Local variables](#) ^[312]

3.11.3.2 IsLinkedToAPI function

Function IsLinkedToAPI (AAddress: to point): Boolean ;

This function permits to know if a variable DELPHI that has been joined previously to a variable PLC with the help of the LinkToAPI function is cooled well.

Parameters :

Aaddress : drawer on the variable DELPHI

Result of the function :

The function sends back TRUE if the variable DELPHI is cooled well and FALSE otherwise.

Example :

Bool1: = IsLinkedToAPI (@ X1) ;

3.11.3.3 AllLinksOK function

Function AllLinksOK : Boolean ;

This function permits to know if all DELPHI variable that has been joined previously to variables PLCs with the help of the LinkToAPI function is cooled well.

Parameters :

No parameters.

Result of the function :

The function sends back TRUE if all variable DELPHI are cooled well and FALSE otherwise.

Example :

Bool1: = AllLinksOK;

3.11.3.4 UnLinkToAPI function

Function UnLinkToAPI (Address: to point): Boolean ;

This function permits to undo the tie between a variable DELPHI and a variable PLC.

Parameters :

Address: drawer on the variable DELPHI

Result of the function :

The function sends back TRUE if the tie to been undone and FALSE otherwise.
So no tie exists for the drawer on the variable, the function sends back FALSE.

Example :

Bool1: = UnLinkToAPI (@ X1) ;

3.11.3.5 ExecMacro procedure

ExecMacro procedure (AmacroText: string; AIPAddress: shortstring);

This function permits to execute one distant macro on a station PLC or one macro local Apigraf WIN.

The list of the usable local macros since this procedure is :

WNETSCHANGEFILENAME
 WNETSCOPYFILEREMOTE
 WNETSCREATEDIRECTORY
 WNETSDELETEDIRECTORY
 WNETSREADANDMOVEFILE
 WNETSWRITEANDMOVEFILE
 WNETSREADFILE
 WNETSWRITEFILE
 WNETSSETFILEATTR
 WFCOPY
 WFDELETE
 WFCREATEDIRECTORY
 WFDELETEDIRECTORY
 WFRENAME
 WFSETFILEATTR

The SUPERG macro is not available since this procedure since the application user is executed outside of WIN APIGRAF. However the user can write in DELPHI :

WINEXEC (' "C:\Program Files\OptimaLog\ApigrafIP\Bin\Exe\SUPERGW.EXE" "C:\Program Files\OptimaLog\ApigrafIP\Projets\Test1\Page1.PSG W ",0)

This procedure will execute the supervisory form Apigraf WIN page1.psgw.

Syntax: WINEXEC (C:\...\SUPERGW.EXE "C:\...\Page1.PSGW ",0)

The duplicates quotation mark must be present when the chain contains some spaces.

It is as possible in this case to pass some parameters to the supervisory form :
 WINEXEC ("C:\Program Files\OptimaLog\ApigrafIP\Bin\Exe\SUPERGW.EXE" "C:\Program Files\OptimaLog\ApigrafIP\Projets\Test1\Page1.PSG W 193.0.1.1,0)

In this example all parameters {1} mentioned in the supervisory form will be replaced by 193.0.1.1. (To see [Call of page with passage of parameters](#)^[287])

Parameters :

AmacroText: Text of the macro to execute (Attention: in the case of the distant macros, it is not necessary to specify the IP address in this parameter)

AIPAddress: IP address of the station PLC on which to do the macro in the case of one distant macro. If the macro is local, it is necessary to put an empty chain in this parameter.

Examples :

Execution of a distant macro:

ExecMacro ('SETR100=1200', '193.0.1.1') ;

Execution of a local macro:

ExecMacro ('WNETSWRITEFILE (193.0.1.1, C:\ESSAI.TXT, C:\ESSAI2.TXT)', '')

3.11.3.6 PLCState function

Function PLCState (AIPAddress: Shortstring; ATimeOut: Integer; ProjectName var: Shortstring; SocketState var: Integer): Integer ;

This function permits to know the state of a station PLC, to know if a project is started, if yes, about what project it is, so Apigraf is started or if the station is on line of order.

Parameters :

AIPAddress: IP address of the station of which one wants to know the state (shortstring)

AtimeOut: Specify the value of time out in milliseconds in the case or the station doesn't answer.
 (Counseled value: 100)

ProjectName var: if a project is started on the station PLC, this chain of type shortstring will contain the name of the project in progress (example: C:\APIGRAF\EXEMPLES\CHENILL2\CHENILL2.EXE)

SocketState var: this variable contains the state of the socket and can be useful if the function in time out doesn't send back the state of the station PLC. The possible values are :

- 0: Closed
- 1: Listening
- 2: SynSent
- 3: SynReceived
- 4: Established
- 5: FinWait1
- 6: FinWait2
- 7: CloseWait
- 8: Closing
- 9: LastAck
- 10: TimeWait

In the case or the station cannot be joins on the network ethernet, the value of the state of the socket will be 2 (SynSent).

Result of the function :

The function sends back an Integer type.

The possible values are :

- 0: State of the station Unknown PLC
- 3: A Project is started
- 6: The workshop Apigraf PLC is started on the station but no project is thrown
- 7: The station PLC is on line of order with INITAPI

Example :

Var Pname: shortstring ;

SockState: Integer ;

EtatPLC: = PLCState ('193.0.1.1' ,100, Pname, SockState) ;

3.12 Variables local OPTIMA VIEW

3.12.1 General principles

OPTIMA VIEW possess global local variables. These variables are recorded to the level of the server of SERVAPIOPC.EXE data

These variables are therefore global to all projects using the server of data.

The state of these variables is recorded in a file text: SERVAPIOPC.VAR and these variables are reloaded since this file to every starting of the server of data.

The local variables are accessible since the supervisory forms Optima View as well as since the programs user.

3.12.2 Description of the local variables

The maximal number of every type of variable can be defined to the level of the server of SERVAPIOPC.EXE data. It is sufficient to open the server in duplicate clicking on its icon in the left corner of the screen and to choose the menu "Opted - > Constants systems"

WR registers

The WR registers are of integer type on 4 Bytes. They are accessible since the supervisory with the WRO expressions, WR1, .WR100,

Registers real WRR

The WRR registers are of type real extended on 10 Bytes. They are accessible since the supervisory with the WRR0 expressions, WRR1, .WRR100.

Flags logical WL

The WL flags is of boolean type. They are accessible since the supervisory with the WL0 expressions, WL1, .WL100.

Switches logical WSA

The switches logical WSA are of boolean type. They are accessible since the supervisory with the WSA0 expressions, WSA1, .WSA100.

Characters strings WATER-CLOSETS

The chains WATER-CLOSETS are of type string and are accessible since the supervisory with the WC1 expressions, WC2,.. WC100.

3.12.3 Definition of the number of local variables

The maximal number of every type of variable can be defined to the level of the server of SERVAPIOPC.EXE data.

It is sufficient to open the server of SERVAPIOPC.EXE data in duplicate clicking on its icon in the left corner of the screen and to choose the menu "Opted - > Constants systems" and to redefine NbMaxWR, NbMaxWRR, NbMaxWSA, NbMaxWL, NbMaxWC.

3.12.4 Visualization of the local variables

To visualize the local variables it is necessary to open the server of SERVAPIOPC.EXE data in duplicate clicking on its icon in the left corner of the screen and to open the menu "Visualization - > local Variables" then to choose the type of variable to visualize in the submenu: WR, WRR, WSA, WL or WATER-CLOSETS.

3.13 OPC server: ServApiOPC

3.13.1 Generalities

The OPC server of optima VIEW is contained in the ServAPIOPC.EXE application. This application automatically rushes as soon as one tempts to reach the server's data. In the same way, the server automatically closes itself when he is not used anymore. The main window of the application servapiOPC can be open when this one is thrown in the administrator of services APIGRAF IP. It is necessary to make a double click on the IP icon in the right lower corner of the screen.

The OPC server answers the OPC requirements: DATES ACCESS V2.0.

A client external OPC can reach the data of the OPC server.

All variables PLCs of apigraf PLC, all variables users recorded on the part PLC with the help of "ENREGISTREMOT " as well as all local variables are accessible with the help of the OPC server.

These variables are by default accessible in only reading (Read only) for reasons of security. However the user

can define the list of the variables that he wishes to consult in read/write (Read / Write).

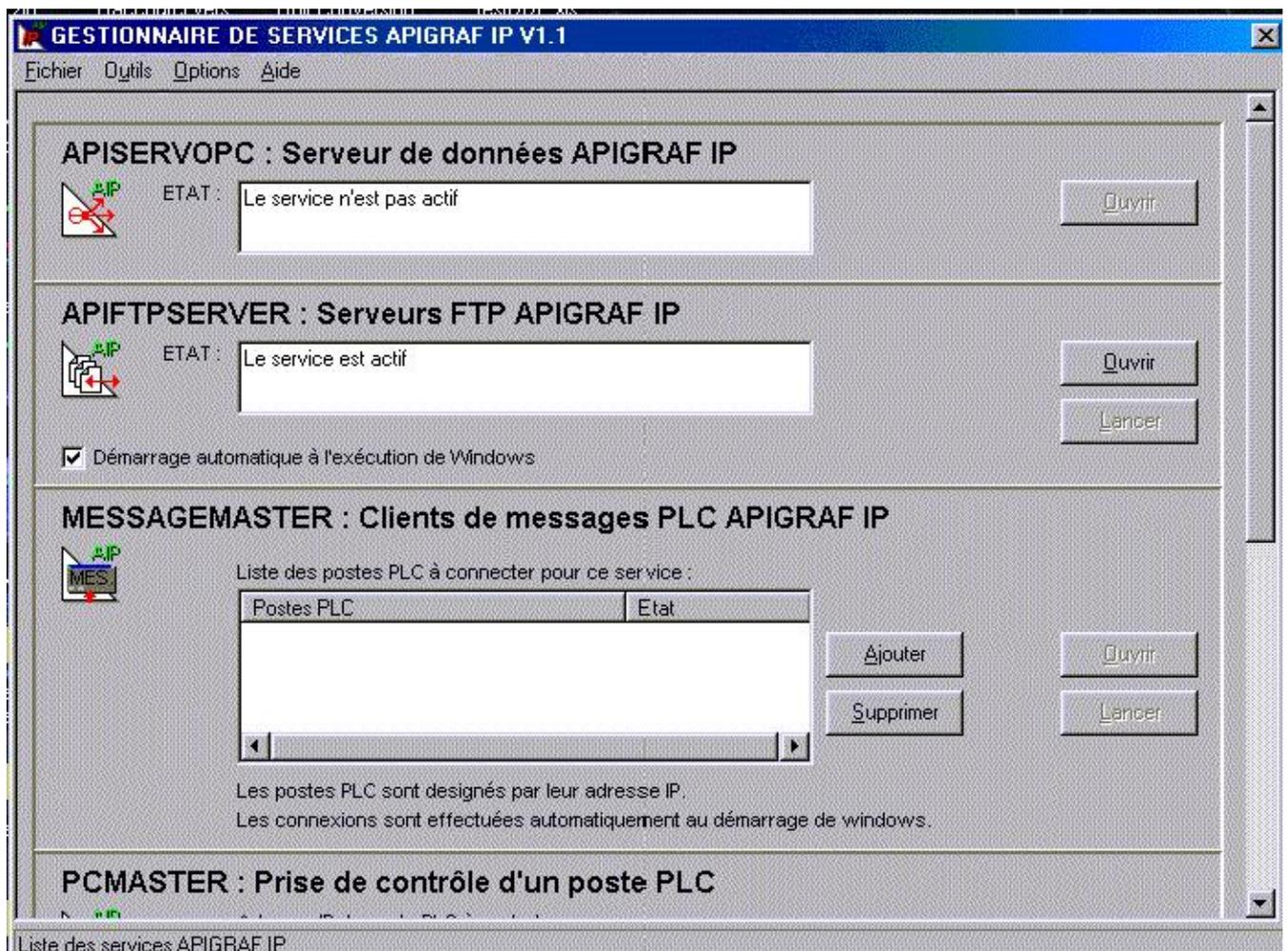
3.13.2 ServApiOPC application

The main window of the application servapiOPC can be open when this one is thrown in the administrator of services APIGRAF IP. The opening of the OPC server makes itself since the menu of Optima PLC. If the supervisory workshop functions regardless of Optima PLC, to follow the following instructions to open the OPC server :

To make a double click on the IP icon in the bar of the tasks to quoted of the clock :

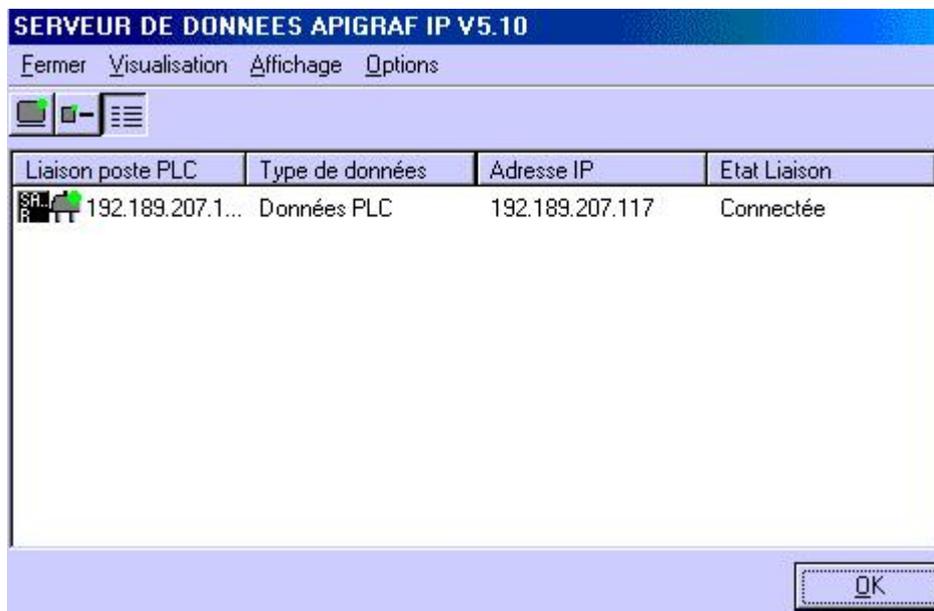


The window of the service administrator opens up :



The APISERVOPC service is in top of the page, if this one is active, one can open it by clicking on the button " TO OPEN ".

The main window of the OPC server opens up then to the screen :



The window main watch the list of connected them with stations PLC. The stations PLC are represented by their IP address. The state of the link is also displayed. It is possible to make a double click on one of the stations PLC to see all variables that are supervised currently on this station PLC.

3.13.3 Access to the data of the OPC server

3.13.3.1 Name of the OPC server

The name of the OPC server for the access to the data is: APIGRAFIP.PLC

3.13.3.2 Definition of a group

Only some private groups can be created on the OPC server.

3.13.3.3 Definition of an item

The items designate in fact the data to which one wants to reach via the OPC server. The valid names for an item are composed of the IP address of the station PLC on which one wants to supervise the variable as well as of the name of the variable to supervise :

Examples :

Variable R100 to the address 193.0.1.1
Name of the item: !193.0.1.1 !R100

R300+R200 expression to the address 193.0.1.1
Name of the item: !193.0.1.1 !R300+R200

Variable local WSA10
Name of the item: WSA10

3.13.3.4 Definition of the data in read/write

By default, all accessible variables since the OPC server are in only reading for reasons of security. However, the user can define a list of variables in read/write.

For it, it is necessary to open the main window of SERVAPIOPC. Then, it is necessary to choose the menu " visualization " then "variable read / write for OPC clients"
The following window opens up then :

Variable	Adresse IP	Variable Locale
R100	192.189.207.117	Non
R200..300	192.189.207.117	Non
WR100..200		Oui
		Oui
		Oui

Buttons: Supprimer, Ajouter, Fermer

This window shows the list of the variables defined in read / write. To add some variables in read / write, to click on the button "to Add." The following window opens up :

Nom de la variable :	<input type="text"/>
Adresse IP :	<input type="text"/>
(Le champ "Adresse IP" doit être vide pour les variables locales)	
<input type="button" value="Annuler"/> <input type="button" value="OK"/>	

The names of variables can be "beaches of variables":

Example : R100.. 200

In the case of the local variables, it made a mistake to leave empty the field "IP addresses"

3.14 Use of the workshop in independent mode

The supervisory workshop can be in installed some cases and can start without OPTIMA PLC. In this case, the software possesses some supplementary functions as for example the selection of the work project.

3.14.1 Installation and starting of the application of Supervisory

3.14.1.1 Installation of the application

The complete installation takes place on two stations (at least).

The station 1 is called Station APIGRAF PLC.

The station 2 is called Station of WINDOWS supervisory.

Installation of the kernel APIGRAF PLC on the station 1

To start the station.

The APIGRAF station must be fit imperatively with an Ethernet board. The launching of the " packetdriver " (driver associated to the board network) must be done automatically to the starting of the station.
To insert the first disk of installation of the kernel APIGRAF PLC in the reader of disk.

To hit " TO: INSTALL "

To follow the instructions of the utilitarian of installation.

To start again the PC.

To hit " APIGRAF " to throw the kernel APIGRAF PLC.

At the time of the first launching, the programs asks you to configure the IP address of the station. Choose the IP address of the kernel APIGRAF PLC then.

Attention : The IP addresses of the station of the Kernel APIGRAF PLC and the station of SUPERVISORY must be compatible with the definite masks of under-network on the two stations.

Example : If the mask of coins definite network on the 2 stations is 255.255.255.0, the compatible addresses are the addresses whose first 3 numbers are equal: 193.0.1.1 and 193.0.1.32 for example.

The graphic supervisory from afar some station APIGRAF PLC is only possible if an APIGRAF project is under execution on the station. On the other hand, the hold of control of the station APIGRAF PLC is at all times possible with the help of the utilitarian PCMASTER (Icon " APIGRAF Distant PLC " on the office). This utilitarian allows you to develop and to execute some projects on the station APIGRAF PLC since the station of supervisory.

Installation of the supervisory tools under Windows

The station under windows® (95-98 or 2000, NT) must be fit imperatively with an Ethernet board appropriately configured in the system.

To insure that the configuration network of the station supports the TCP / IP protocol well and that an IP address has been defined by hand in the system :

For it, to stand in " MENU TO START > PARAMETERS > PANEL OF CONFIGURATION > NETWORK "

To verify the presence of an attached TCP / IP protocol to the Ethernet board, it generally results in the presence of a line of type : " TCP / IP - > " Name of the Ethernet board " ". So such is the case, to pass directly to the point 6.

If it is not the case, to choose " TO ADD ", to choose then " PROTOCOL " , then " MICROSOFT " and finally " TCP / IP " .

To validate with " OK " .

In the list of the components network of the window network, to choose with the help of the mouse the line corresponding to the protocol " TCP / IP - > Board Ethernet " . The line appears then in surbrillance.

To click then on the button " PROPERTIES " .

In the tab " IP ADDRESS " , to check " To specify an IP address " . To choose the IP address of the station and the mask of under-network then.

To validate the windows with " OK " . If the system asks to start again, to start again the station before pursuing the installation.

To insert the first disk of installation of the APIGRAF supervisory tools under windows®. (To insert the CD if the support is a CD)

In a window " explorer windows® " , to select the reader of disk and to execute " SETUP.EXE " .

To follow the instructions of the utilitarian of installation.

Note : The installation of the kernel APIGRAF PLC can be done on several stations, knowing that the station of WINDOWS® supervisory can supervise several kernel APIGRAF PLC.

3.14.1.2 Starting of the application

To double-click on the abridgment OPTIMA VIEW on the office of the supervisory station to throw the supervisory workshop.

A window of project selection permits to choose an existing project or to create a new of it.

At the time of the first launching, to stand in the directory " PROJECTS " and to choose "to create a new project "

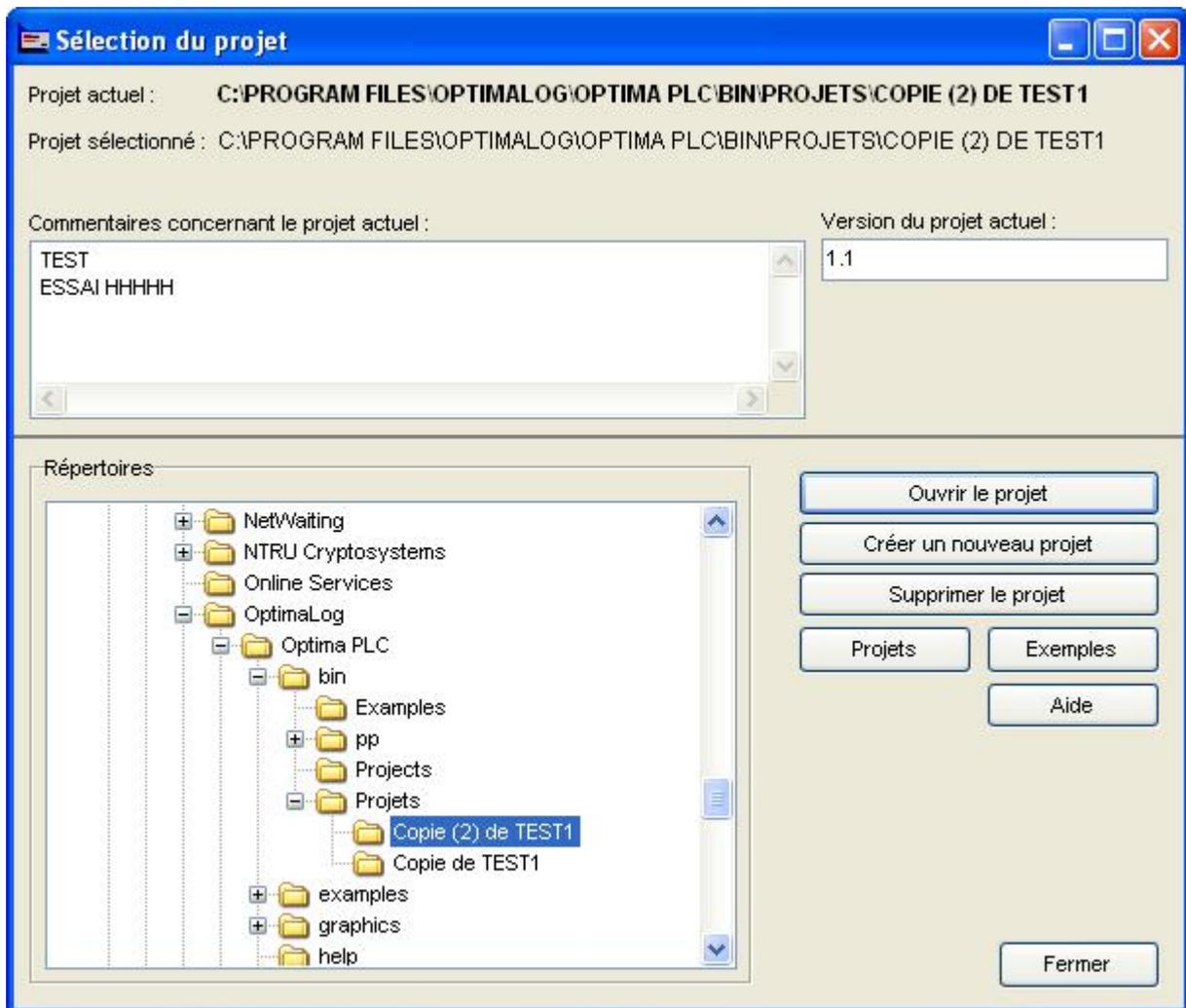
Advanced users :

Once some supervisory forms have been created, it is possible to use a line of order (in an abridgment or since another application) to execute a supervisory form, the line of order will have the following shape then :

```
"C:\Program Files\OptimaLog\Optima View\Bin\Exe\SUPERGW.EXE" "C:\Program Files\Optimalog\Optima View\Projets\TEST\Page1.PSG W"
```

It is also possible to pass some parameters to the supervisory form to open :

```
"C:\Program Files\OptimaLog\Optima View\Bin\Exe\SUPERGW.EXE" "C:\Program Files\OptimaLog\Optima View\Projets\TEST\Page1.PSG W 193.0.1.1"
(To see Call of page with passage of parameters287)
```

3.14.2 Window of project selection**3.14.2.1 Selection of a project**

The window of project selection displays the information in progress concerning the project (Name of the project, commentaries, version,...). It appears to every starting of the application.

Its dialog box type permits to select the directory of the project that one wants to open as well as the reader on which one wants to work.

The window presents the following controls :

Button " To open the project "

Open the project contained in the directory in subbrillance in the window of exploration of the projects.

Button " To create a new project "

Create a new project in one coins directory of the directory in subbrillance in the window of exploration of the

projects.

Attention: The creation of projects is only allowed in the directory " PROJECTS " or in one of the coins directories of " PROJECTS "

If the current directory is not the PROJECTS directory, a message of confirmation asks if one wants to create the project in the PROJECTS directory. To answer "yes" to the message of confirmation then to enter the name of the project to create.

Button " To suppress the project "

Suppress the project in surbrillance in the window of exploration of the projects. The current project cannot be suppressed.

Button " To close "

Farm the window of selection of the projects.

Button " Projects "

Open the directory of the projects.

Button " Examples "

Open the directory of the examples.

4 Runtime Optima PLC

The execution of an application developed with the development workshop Optima PLC implies the use of the runtime Optima PLC.

Although an man/machine interface is often sufficient for the management of a system, the environment of the runtime offers possibilities of visualization and modification of the variables of the application.

4.1 Generalities

Once developed and compiled in the workshop Optima PLC, an application is constituted mainly of an OptimaPLC.prg file situated in the directory of the project and containing the code to execute.

The "runtime" Optima PLC is a program common to all Optima PLC applications, permitting the execution of these. It is not an interpreter, but an interface between the code and Windows on one hand, between the code and the user on the other hand.

The runtime allows the user to visualize and to modify the internal variables of the application, to follow the progress of the PLC program, and to assure the debug and the fixing of it. It doesn't constitute a convivial interface for the daily exploitation by operators, but gives access to important information for the developer and the repairman.

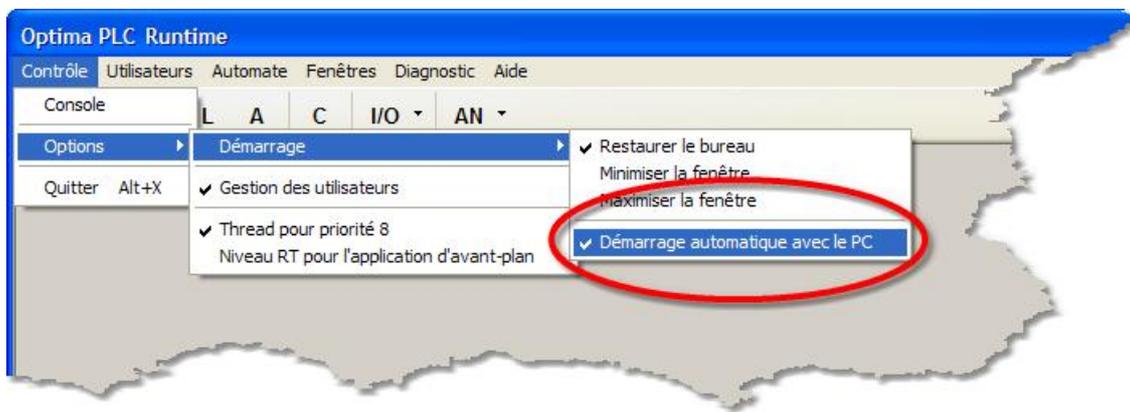
4.1.1 Starting of the application

The application Optima PLC, once compiled, presents itself as an OptimaPLC.prg file situated in the directory of the project. Several possibilities exist to run it :

- Direct launching from the workshop Optima PLC: it is the simplest method during the phase of development.
- Manual launching from the exploring Windows: double-click on OptimaPLC.prg to start the execution.



- Automatic launching to the starting of the PC, defined by hand: create a shortcut for the OptimaPLC.prg file (contextual menu by right click), and place this one in the directory "Starting" (Start Menu / Programs / Starting).
- Automatic launching to the starting of the PC, defined from the runtime: use the menu Controls / Options / automatic Starting / Starting with the PC.



Warning : under Windows Vista or Windows Seven, to allow a normal start of the runtime, it may be necessary to disable the UAC (user account controller), which might otherwise obstruct the automatic startup of software that requires high access rights to equipment such as Optima PLC (disabling this can be done in the Control Panel, User Account Control).

4.1.2 Stop of the application

The closing of the runtime window doesn't stop this one, but only minimizes it only. Its icon remains present in the Windows taskbar :



To stop the runtime completely, several possibilities exist :

- In the workshop Optima PLC, when the runtime has been launched from the workshop, click on the button .
- In the runtime, use the menu Controls / Quit.
- Double-click on the icon in the Windows taskbar : in the menu that appears, click on Quit.



- In the case of closing of Windows or Windows session, the runtime is automatically stopped (do not force the stop immediately if it is proposed by Windows).

Please note that a demand of stop by the user is always followed of a confirmation demand before the complete stop.

A stop provokes the passage of the application in phase STOP, during which some tasks programmed can do a finalization more or less long. The runtime stops the hardware drivers, what can also take a little time, especially if a problem of communication is met for example.

4.1.3 Users management

It is possible to restrict the possibilities of action in the runtime according to the user. For it, the mode "management of the users" must be activated in the menu Controls / Options / Management of the users.

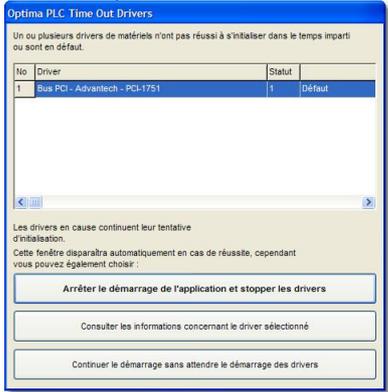
The activation of this mode provokes the apparition of the menu "Users" in the menu bar.

For more of information, please refer to the chapter [Users management](#) ²⁸.

4.1.4 Status of the application

The state in which is the application is displayed in the statusbar.

The different states are :

Phase		Interpretation
1	START DRIVERS	Before starting the PLC program, all drivers must be initialized.
2	START DRIVERS TIMEOUT	<p>If on (or several) driver(s) cannot be initialized, a list of the drivers non initialized is displayed in order to warn the user. The attempts of initialization continue until the driver initializes itself correctly, or the user intervenes.</p> 
3	START PLC	The tasks having to run in this phase of initialization of the PLC program are executed. The phase ends when all these tasks toppled to 1 their variable StartPLCPhaseOK.
4	RUN	The application is under normal execution.
5	FAULT	The application is under execution but one at least of the material elements bound to the application (hardware of input / output,...) don't function as foreseen.
6	STOP PLC	The tasks having to be executed in this phase of finalization of the PLC program are executed. The phase ends when all these tasks toppled to 1 their variable StopPLCPhaseOK.
7	DRIVERS STOP	Before stopping the runtime, all drivers must be stopped.
8	STOP DRIVERS TIMEOUT	If one (or several) driver(s) cannot be stopped, a list of the drivers non stopped is displayed in order to warn the user. The attempts of stop continue until the driver stops correctly, or that the user intervenes.

4.1.5 Performances and real-time

The general performances of the application depend on the following options :

- Activation of the real-time kernel (General Options / Compilation menu of the workshop)

This option permits to activate or no the real-time kernel.

If it is active, the levels of priority 0 to 7 are managed by this kernel, that is priority over other action of Windows. The performances are optimal. On the other hand it is possible to meet problems or conflict of resource with this mode, so it is therefore counseled to use it only on a PC dedicated to the application.

If it is inactive, the levels of priority 0 to 9 are all managed by the multitask preemptive kernel of Windows.

- Windows tasks of elevated priority ([Control / Options menu of the runtime](#)^[324])

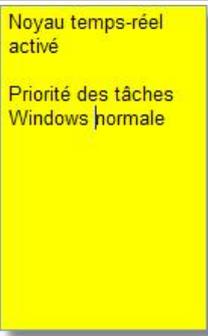
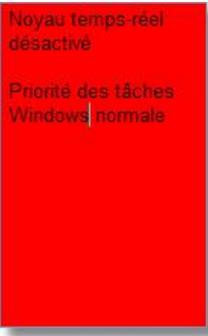
This option has the effect of giving or no a raised priority to the application.

If an elevated priority is given, all parts of the application depending on Windows will be priority over the others application and the system tasks of Windows. If the real-time kernel is deactivated, this option is of as much more important.

If the normal priority is left, the application is executed in a classic manner, with periods for the PLCs tasks impossible to master.

To visualize the level of the performances, use in the runtime the [Diagnosis / Performances menu](#)^[335]. This screen permits to visualize the key values of timing.

The displayed on the right panels have the following significance :

 <p>Noyau temps-réel activé Priorité des tâches Windows élevée</p>	<p>Optimal performances, real-time kernel activated , tasks Windows priority.</p>
 <p>Noyau temps-réel activé Priorité des tâches Windows normale</p>	<p>Optimal performances of the real-time tasks, but the tasks managed by Windows are not optimized (levels 8 and 9).</p>
 <p>Noyau temps-réel désactivé Priorité des tâches Windows élevée</p>	<p>All tasks are managed in a priority manner by Windows, what permits to get correct performances for an application of classic automatic device.</p>
 <p>Noyau temps-réel désactivé Priorité des tâches Windows normale</p>	<p>All tasks are managed by Windows to the same level of priority (or nearly) that the other applications, what usually doesn't permit to make operate an application correctly (safe for simulation, teaching, etc...)</p>

4.2 Description of the environment

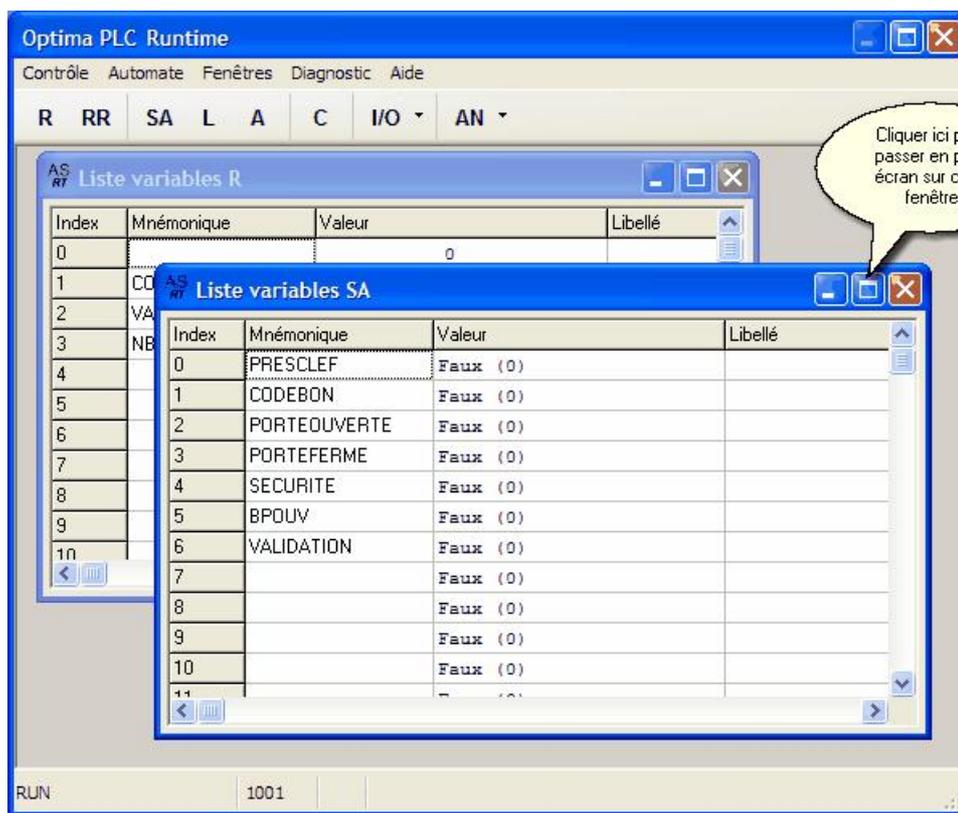
4.2.1 Main window

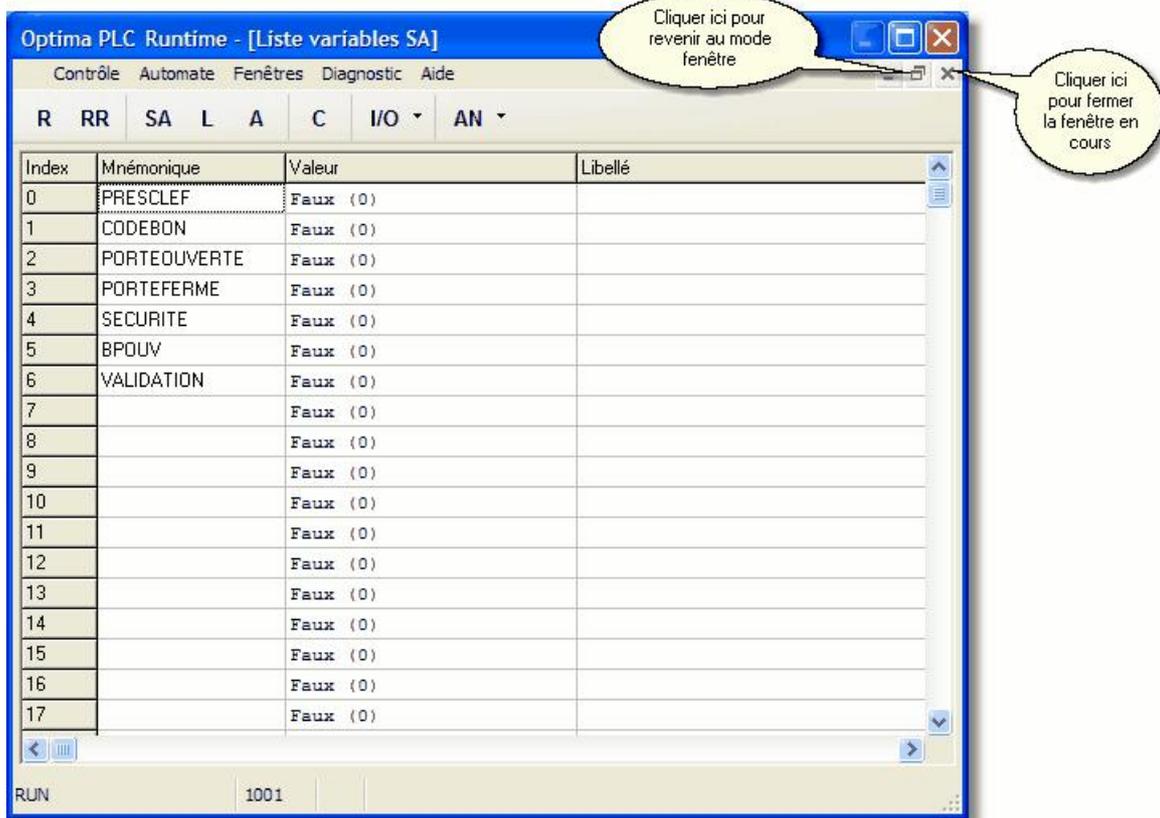
The environment of the runtime is composed of 4 parts :



- the menu (at the top of the window)
- the toolbar (under the menu): it is composed of buttons permitting the display of the different PLC variable of Optima PLC
- the editor (in the center of the window): to open the visualization windows
- the statusbar (at the bottom of the window) that indicates the state of the application

The different visualization windows or control opened in the runtime Optima PLC are included in the main window, and can be zoomed in according to different modes :



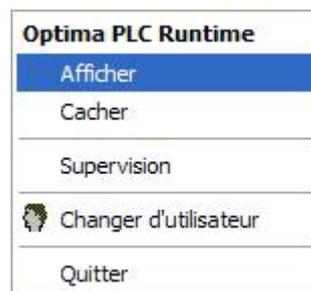


4.2.2 Recall of the window

After one period of inactivity of the user, the window is automatically reduced. To bring it back to the screen, it is necessary to do a double-click on the icon situated in the statusbar of Windows :



The menu that appears permits réafficher the window of the runtime :

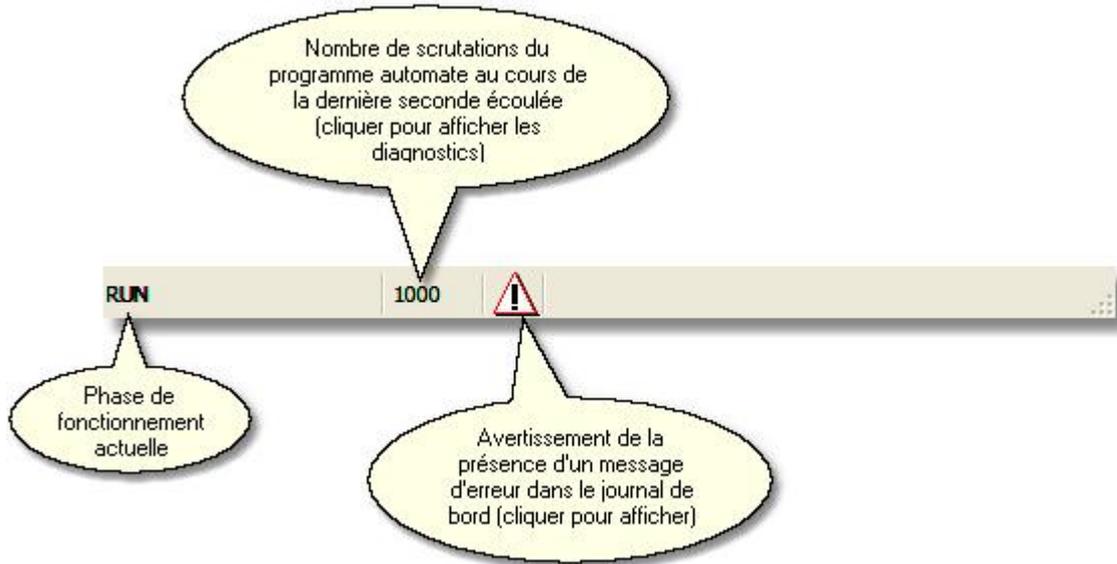


If the management of the users is activated, it is necessary to enter his/her/its identifying and password :



4.2.3 Statusbar

The statusbar placed to the low of the window of the runtime provides some information :



To see the paragraph "[Statute of the application](#)" for more of information concerning the phase of working.

4.3 Description of the menus

4.3.1 Menu Control

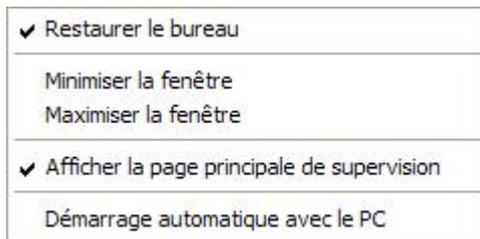
This menu permits to modify the options of working of the runtime and to leave the application.

4.3.1.1 Menu Options

This menu gives access to some options of working of the runtime.

- Starting

This option gives access to a new menu :



Option	Effect
Restore desktop	<p>The aspect of the desktop is memorized at the time of the output of the application to be restored to the following starting. All windows are not concerned however by this restitution.</p> <p>In mode "users management", the desktop is restored according to the user.</p> <p>The files containing the description of the desktop are situated in the Desk subdirectory of the project, and their name is constituted of the name of the user followed of the .dsk extension.</p>

Minimize the window	At starting, the window of the runtime is minimized.
Maximize the window	At starting, the window of the runtime is maximized.
Display the main page of supervisory	The main supervisory form (PSGW file with the same name that the project) is displayed at starting.
Automatic starting with the PC	<p>At the time of the PC starting, the runtime automatically starts with this project.</p> <p>If another project Optima PLC was configured in automatic starting, the fact to check this option replaces the starting of the previous project by the starting of the current project.</p> <p>When the option is unchecked, no project Optima PLC is started automatically.</p> <p>Warning : under Windows Vista or Windows Seven, to allow a normal start of the runtime, it may be necessary to disable the UAC (user account controller), which might otherwise obstruct the automatic startup of software that requires high access rights to equipment such as Optima PLC (disabling this can be done in the Control Panel, User Account Control).</p>

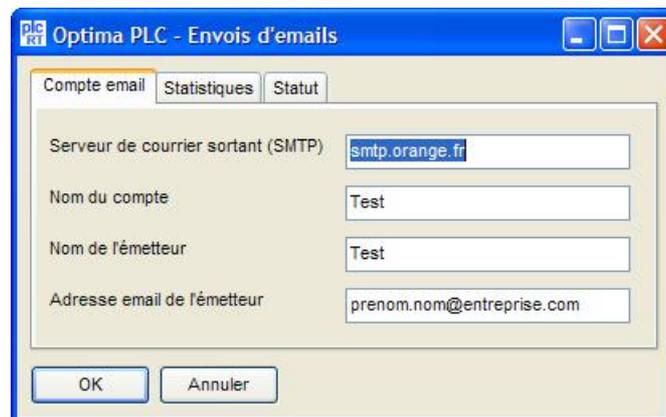
- Users management

This option permits to choose the mode of users management, in which an identification is asked previously for the access to the screens of the runtime. This mode offers a better security in a multi-users environment, what is often the case on a production machine.

When this mode is activated, the rights attached to the user restrict him for the access to some functions and menus. Besides, a special menu of users management is displayed.

- Sending of email

This option permits to parameterize and to control the integrated function of emails sending.



The tab "Email account" must be configured correctly with parameters of connection to an email server (SMTP) sending functional.

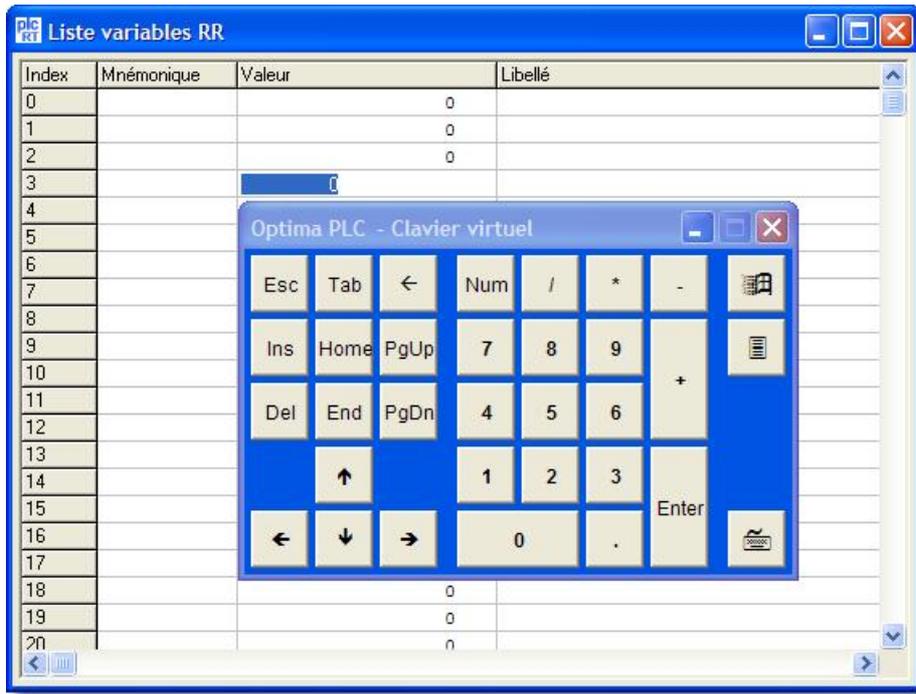
The tab statistics indicates the number of emails emitted. Finally, the Status tab displays the current state of the emails sending.

- Automatic virtual keyboard

This option, if it is checked, indicate that the virtual keyboard of Optima PLC must be displayed as soon as an input is proposed to the user. It is interesting if the application is executed on a PC with a tactile screen and without keyboard. The screen is used then like input keyboard.

If the input zone changes or is displaced, the virtual keyboard is also displaced. When there is no more input in

progress, the keyboard is automatically hidden.



Above, in the example the beginning of input of the RR3 register provokes the display of the virtual keyboard. The window of this one is placed closer to the input zone.

- Windows tasks of elevated priority

This option permits to give to the Windows part of the application the maximal priority (Real Time). It is counseled to improve the general performances of the application, in particular those non based on the integrated real time kernel, but on Windows.

- Period for tasks of priority 8

This submenu, visible solely if the real-time kernel is activated, permits to define the period of execution of the tasks of priority 8. For recall, this level of priority is not real time, but managed by Windows. The period is by default of 10 ms.

- Submenu "real Time"

This submenu permits to parameterize tolerance on the basis period of the cyclic treatments. If the period noted fate of tolerance, a warning message is displayed. Tolerance is defined by a percentage applied on the period. It is counseled to do some tests before fixing a too low value.

- Submenu "Data retain"

The data retain are saved periodically in disk files. This submenu permits to adjust the back-up interval (in seconds) and the number of files kept.

4.3.1.2 Option Licenses

This option permits to display the current information concerning the user's license of the software.

Optima PLC - Licences d'utilisation

Optima PLC

Localisation

Aucune clé

Clé USB locale

Clé USB distante

PC

No de série 105277

Type de licence Gold

Numéro de licence 612002

Début/fin de validité 09/12/2006

Nom utilisateur Dupont Automatismes

Code utilisateur 113

Fin de maintenance logicielle

Fin d'assistance technique

Atelier logiciel

Runtime

Mono-application

OPTIMA View

Nb Max I/O TOR

Nb Max I/O Ana

Actualiser

Fermer

Modification Licences

4.3.1.3 Option to Leave

This option provokes the complete stop of the runtime, after a demand of confirmation :

Confirmation

L'application Optima PLC est en cours d'exécution, souhaitez-vous la stopper ?

Oui Non

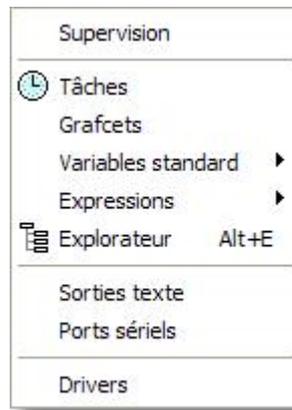
4.3.2 Menu Users

This menu regroups the options permitting to manage the list of the authorized users. It only appears if the mode "users management" has been activated in the options.

The options of this menu are common to the different Optimalog software. For more of information, to see the chapter [Users management](#) ²⁸.

4.3.3 Menu PLC

It is composed of submenus allowing the access to the variables and available various information for the working of the PLC.



4.3.3.1 Option Supervisory

This option provokes the display of the supervisory associated to the application.

If no supervisory form is already displayed, the main form is loaded and displayed. This form corresponds to the file situated in the PSGW subdirectory, with the same name that the project.

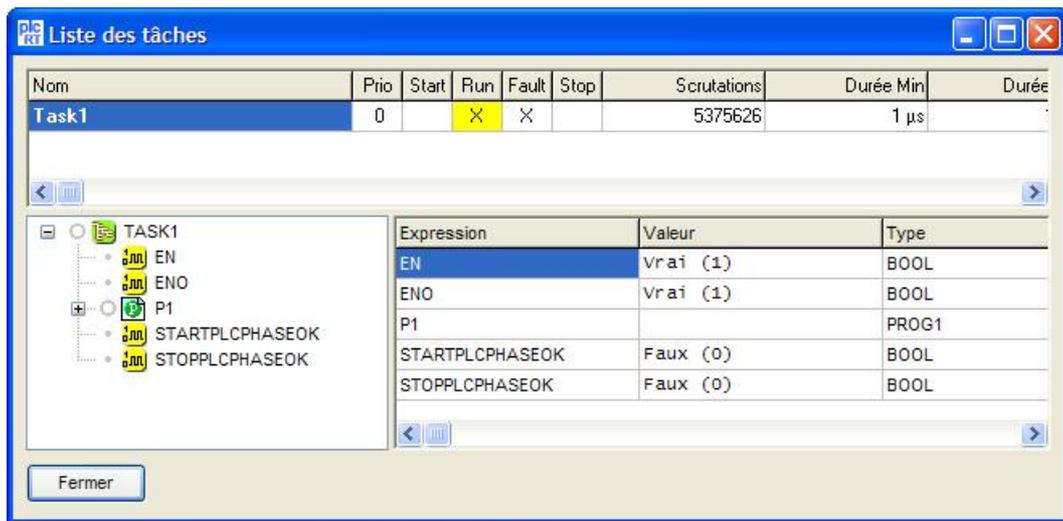
If some pages are already displayed, this option puts them to the first plan.

When some supervisory forms are displayed, the output of the application provokes the closing of the forms.

4.3.3.2 Option Tasks

It is possible with this option to show the list of the tasks of the program.

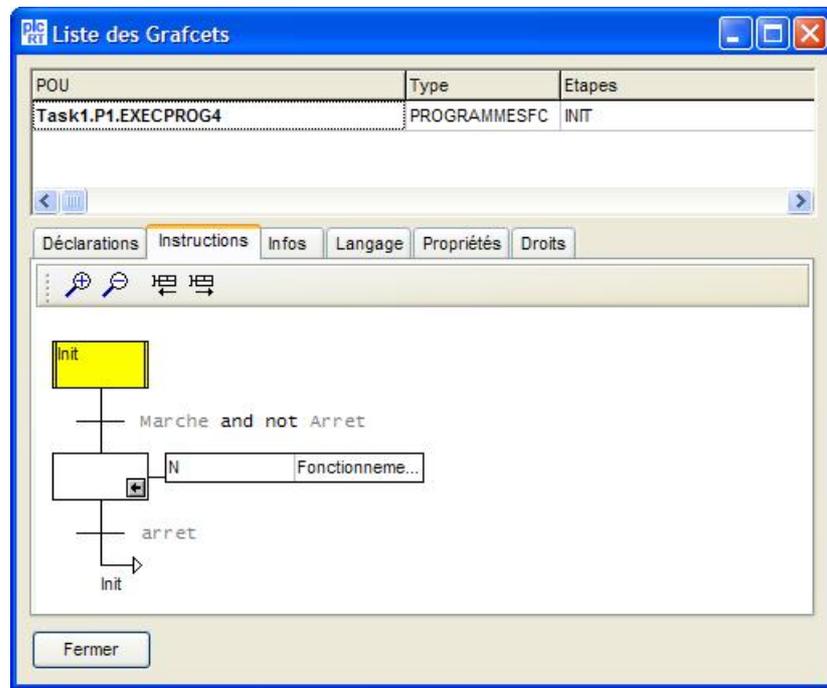
One can visualize different important parameters to every tasks as the minimum and maximum duration of a scrutation, or its priority level in relation to the other tasks of the application.



The selection of a task in the list provokes the display in the low part of the window of a variables explorer based on the execution tree of this task.

4.3.3.3 Graficets option

It is current to begin the debug or the fixing of an automatic device by verifying the sequences of the system and the active steps. The objective of this option is to facilitate the access to these sequences by presenting them all as a list displaying, for each, the active steps.



The selection of a sequence in the list situated in high part high of the window provokes the display of the graph of the sequence in low part. The active steps are displayed in a different color.

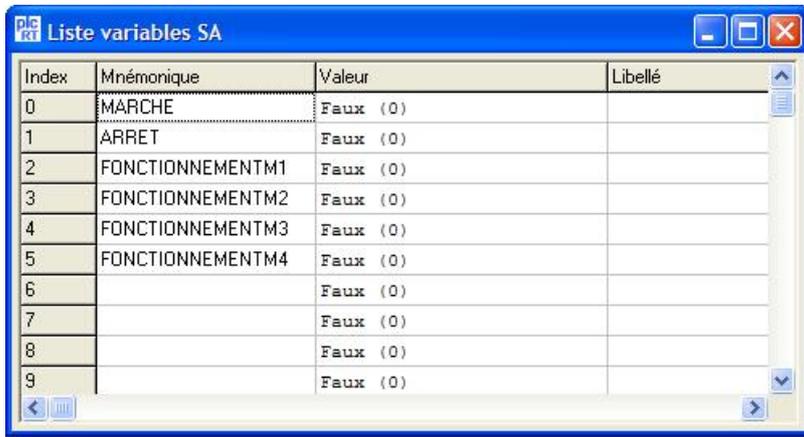
4.3.3.4 Option standard Variables

This menu permits to visualize in real time and sometimes to modify the state of the PLC variables of Optima PLC :

Registres R	Ctrl+R
Registres RR	
Flags SA	
Flags L	Ctrl+L
<hr/>	
Entrées TOR I	Ctrl+I
Sorties TOR O	Ctrl+O
<hr/>	
Entrées analogiques ANI	
Sorties analogiques ANO	
<hr/>	
Chaînes C	

- the R registers
- the RR real registers
- the SA flags
- the L flags (non modifiable)
- the inputs and outputs, digital or analogical (can possibly be forced to a value if the user's rights permit it)
- the C strings

Here is the window corresponding to the SA variables :



Index	Mnémonique	Valeur	Libellé
0	MARCHE	Faux (0)	
1	ARRET	Faux (0)	
2	FONCTIONNEMENTM1	Faux (0)	
3	FONCTIONNEMENTM2	Faux (0)	
4	FONCTIONNEMENTM3	Faux (0)	
5	FONCTIONNEMENTM4	Faux (0)	
6		Faux (0)	
7		Faux (0)	
8		Faux (0)	
9		Faux (0)	

Only the values of the variables can be modified (the mnemonics and commentaries, defined by programming, can only be visualized).

To modify a value, it is necessary to input the new value in the appropriated box, then to validate it with the "Enter" key of the keyboard.

The forcing of the digital inputs/outputs is made by selecting the state to modify then by using the contextual menu that proposes 3 options :

- no forcing
- forcing to 0
- forcing to 1

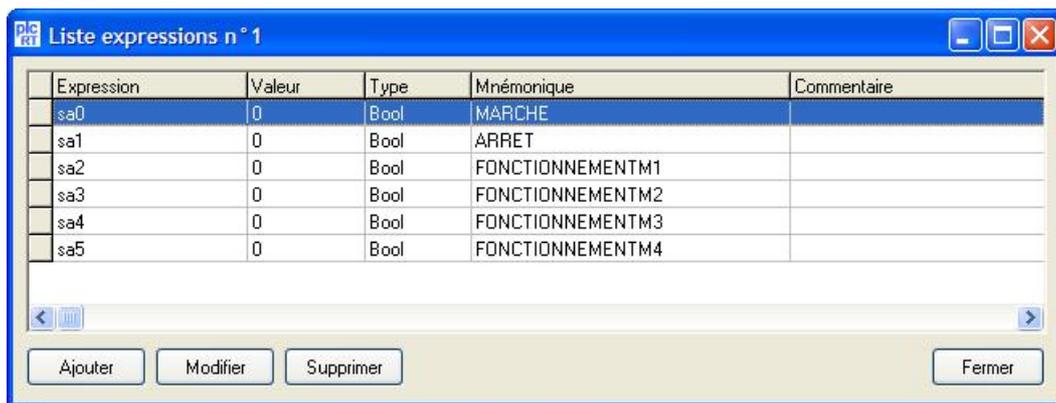
The forcing of the analoical inputs/outputs is made in 2 times :

- the value to affect must be input in the "forcing" column (if this column doesn't appear, it means that you don't have the right to force the values)
- the contextual menu permits then to force the input or the output to this value

4.3.3.5 Option Expressions

An "expression" designates a simple variable or a formula calculated from variables of the application.

The "lists of expression" constitute a visualization tool easily configurable for the user, permitting to regroup in a same window disparate variables but whose simultaneous consultation presents an interest for the debug or the fixing.



Expression	Valeur	Type	Mnémonique	Commentaire
sa0	0	Bool	MARCHE	
sa1	0	Bool	ARRET	
sa2	0	Bool	FONCTIONNEMENTM1	
sa3	0	Bool	FONCTIONNEMENTM2	
sa4	0	Bool	FONCTIONNEMENTM3	
sa5	0	Bool	FONCTIONNEMENTM4	

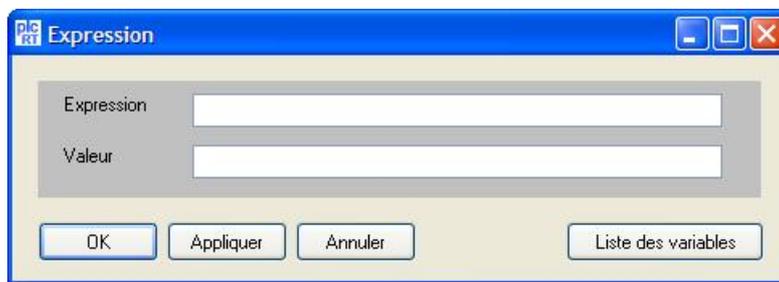
Ajouter Modifier Supprimer Fermer

All preset or variables defined by the user can be used in this tool.

4 lists of expressions can be created.

In each of these lists it is possible to add, to modify or to suppress the display of an expression. When an expression is displayed in a list, its value is updated periodically.

The addition or the modification of an expression makes the following window appear :

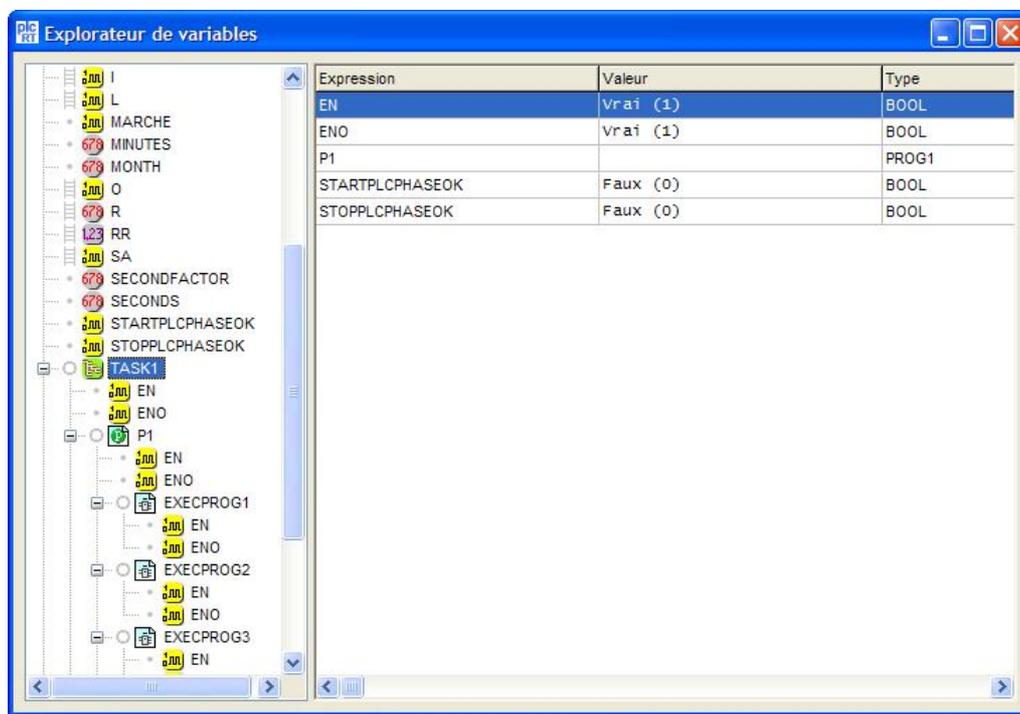


When the name of the variable is modified, the value is automatically updated (provided that the variable exists). When the value is modified, it must be validated (key "Enter" of the keyboard or "OK" button of the window) to be affected to the variable.

An expression can be also created from an variables explorer by dragging the variables with the mouse, and by dropping it on the expressions window.

4.3.3.6 Option Explorer

When one doesn't know the exact name of an expression or when one wishes to consult its value without adding it in one of the lists, it is possible to reach the complete list of the expressions of the application while clicking on the "Explorer" button.

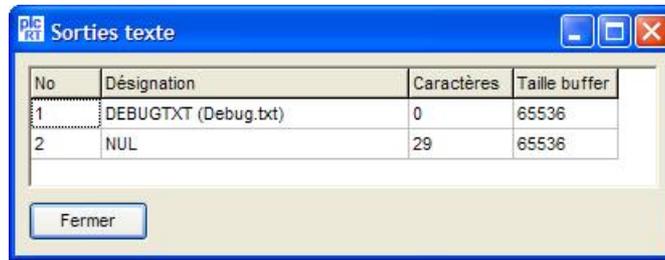


The window that appears is composed of two parts :

- a panel on the left containing the execution tree.
- A list on the right presenting the variables contained in the selected tree level.

4.3.3.7 Option Outputs Text

The "output text files" of Optima PLC permit to define output streams in the application, in which the program can add the text destined to be printed or stored on disk.



For every output are mentioned the number of characters situated in the buffer, as well as the size of this buffer.

Two output text files are always available of basis, NUL (no storage nor impression), and DebutTxt (storage in the Debug.txt file situated in the directory of the project).

- NUL : temporary file automatically created to the starting of the application, its content can be visualized however from the runtime, it is suppressed as soon as the application is stopped
- DEBUGTXT : file automatically created (or open) at the starting of the application, its content can be visualized from the runtime, it is protected then until the stop of the application

To visualize the content of a file, double-click on its name.



A list permits to consult the current content of the buffer, that can possibly evolve quickly according to the use that is made of the output in the program.

To freeze this display and to consult the content of an output frequently used more easily, uncheck "dynamic display",

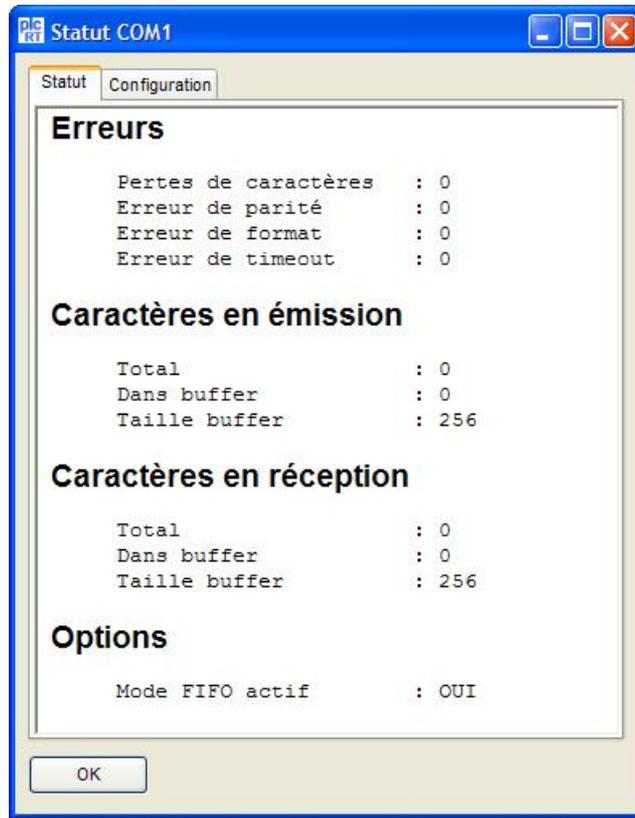
The box "Fixed font" provokes the use of a fixed font instead of a proportional font, what is useful if the output contains some information at constant places on the lines.

4.3.3.8 Option Serial Ports

It is possible to visualize the list of the serial ports used in the application, as well as their parameters (who cannot be modified here).

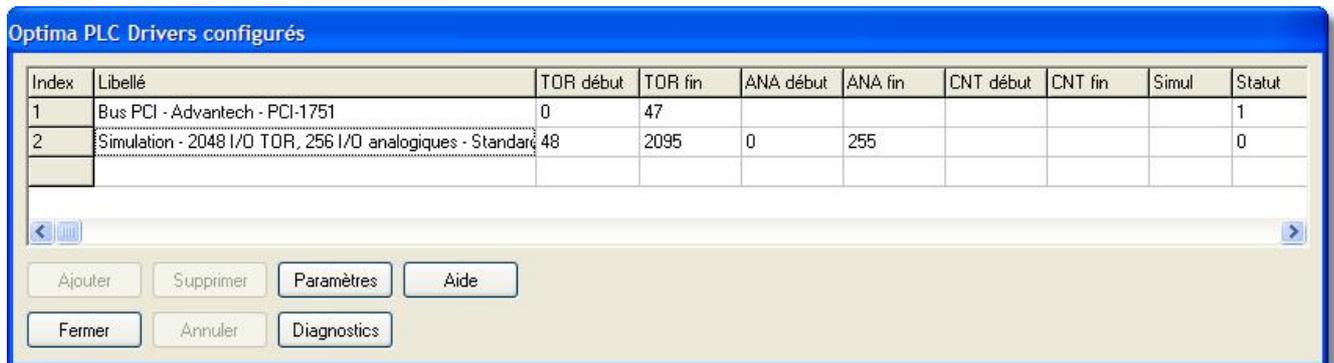


To get more information on a used COM port, double-click it or use the Select button.



4.3.3.9 Drivers option

It is possible here to visualize the list of the drivers used in the application, as well as their parameters.

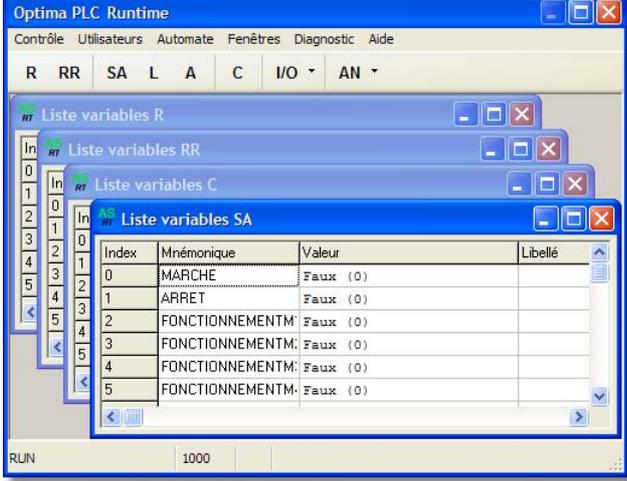


4.3.4 Menu Windows

This menu is dedicated to the management of the windows displayed in the runtime.



Option	Effect
Close all	Close all windows opened in the runtime.

<p>Mosaic</p>	<p>Create a mosaic with the open windows so that all are visible (doesn't concern all windows).</p> 
<p>Cascade</p>	<p>Cascade the display of the windows by partially superimposing them.</p> 
<p>Previous</p>	<p>Activate the previous window in those displayed.</p>
<p>Follow</p>	<p>Activate the following window in those displayed.</p>
<p>Virtual keyboard</p>	<p>Call the virtual keyboard Optima PLC, useful with a tactile screen. This option only appears if the option "automatic virtual Keyboard" is checked in Control/options^[324] (in this case the virtual keyboard is automatically displayed as soon as an input is proposed).</p>
<p>List of the windows</p>	<p>Display a contextual menu permitting to choose among the open windows the one to activate.</p>

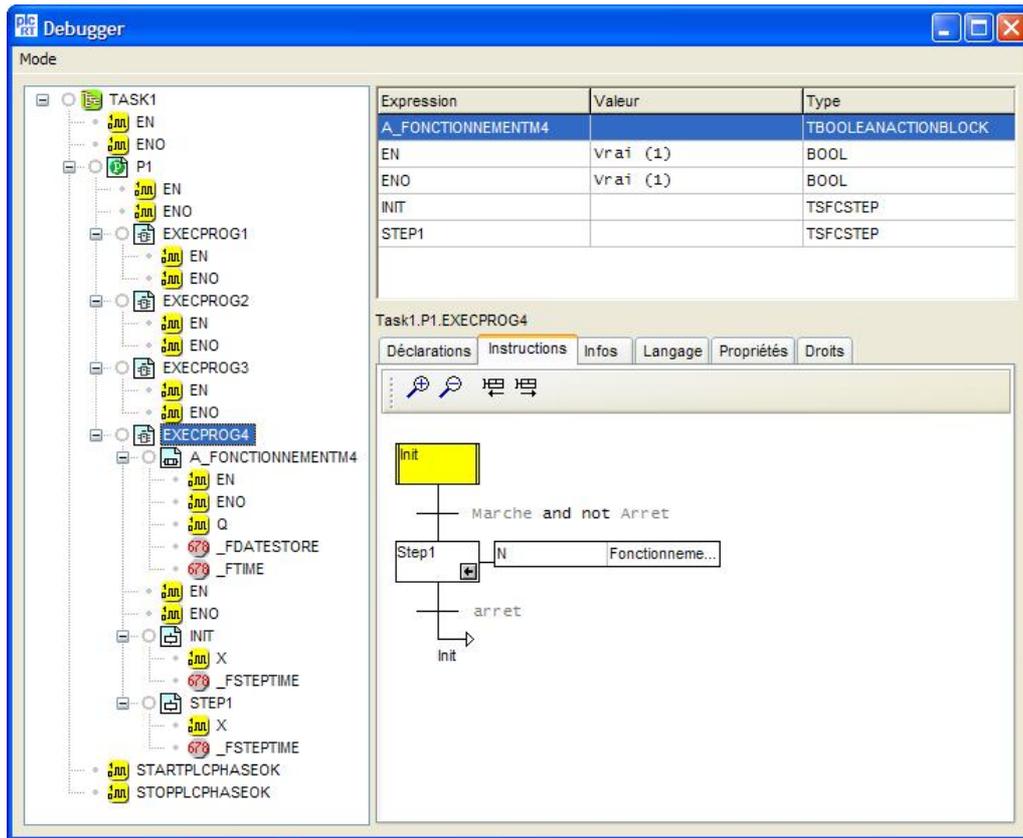
4.3.5 Menu Diagnoses

This menu gives access to various debug functions.

Debugger	Alt+D
Performances	
Journal de bord	Alt+J
Server OPC	
Liste des sockets	
Générer un enregistrement de diagnostic	
Position d'erreur	

4.3.5.1 Debugger option

This option provokes the display of the debugger, that permits a deep follow-up of the working of the program.

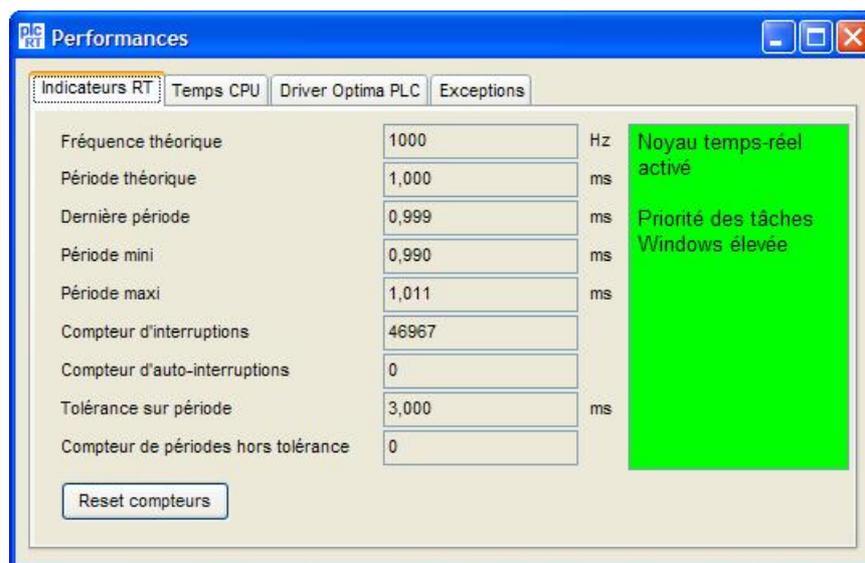


Debugger used with a POU defined in SFC

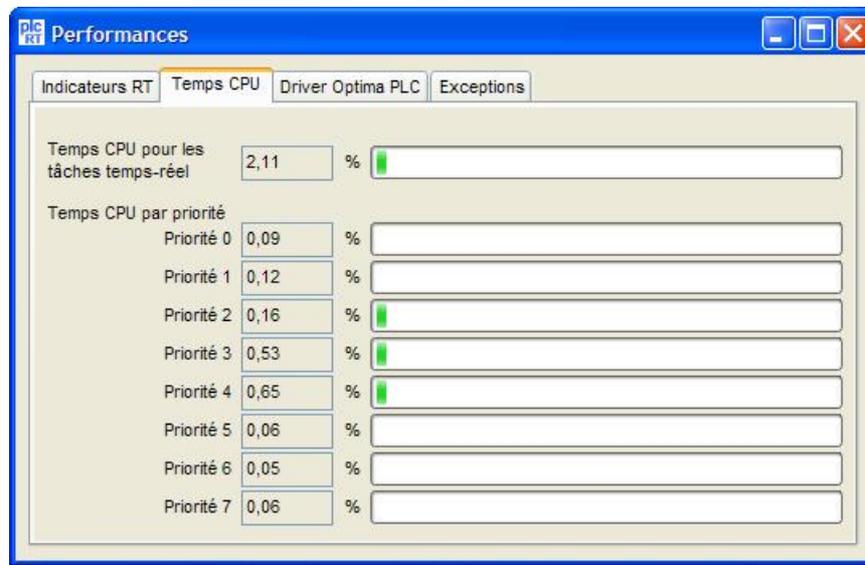
This screen includes a variable explorer as well as a dynamic visualization panel of the POU's selected.

4.3.5.2 Option Performances

The window displayed here presents the information concerning the real-time of the application and on the occupation of the CPU by the PLC program.



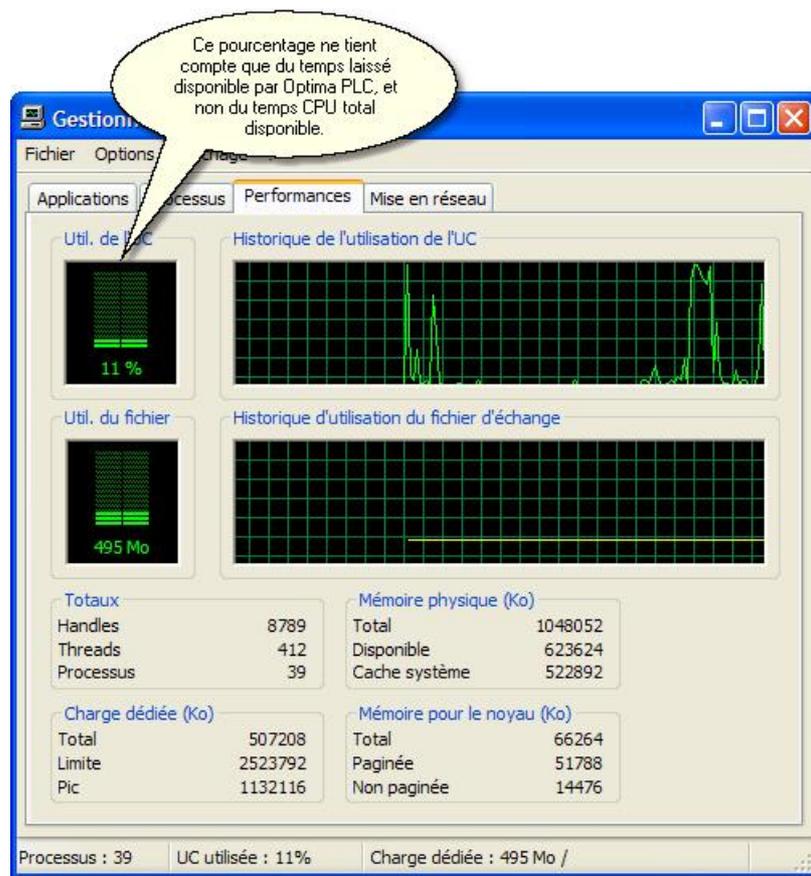
The tab "RT indicators" regroup the main information concerning the real-time. The most important being the strict respect of the theoretical frequency of the execution of the program, the points to supervise are the period maxi and the counter of periods out tolerance, that can drive to diagnose, in case of anomaly, the presence of a hardware or a driver coming to disrupt the real-time functions of Optima PLC.



The tab "CPU Time" permits to control the use of the CPU done by the runtime Optima PLC. The global time consumed by the real-time tasks is displayed globally (in percentage of the available time), the proportion is displayed then for every real-time priority level (0 to 7).

The CPU time consumed by the non real-time tasks is not posted here (levels 8 and 9).

Note that the percentage of the CPU time used, displayed by Windows, corresponds to the percentage calculated in relation to the remaining time non consumed by Optima PLC.



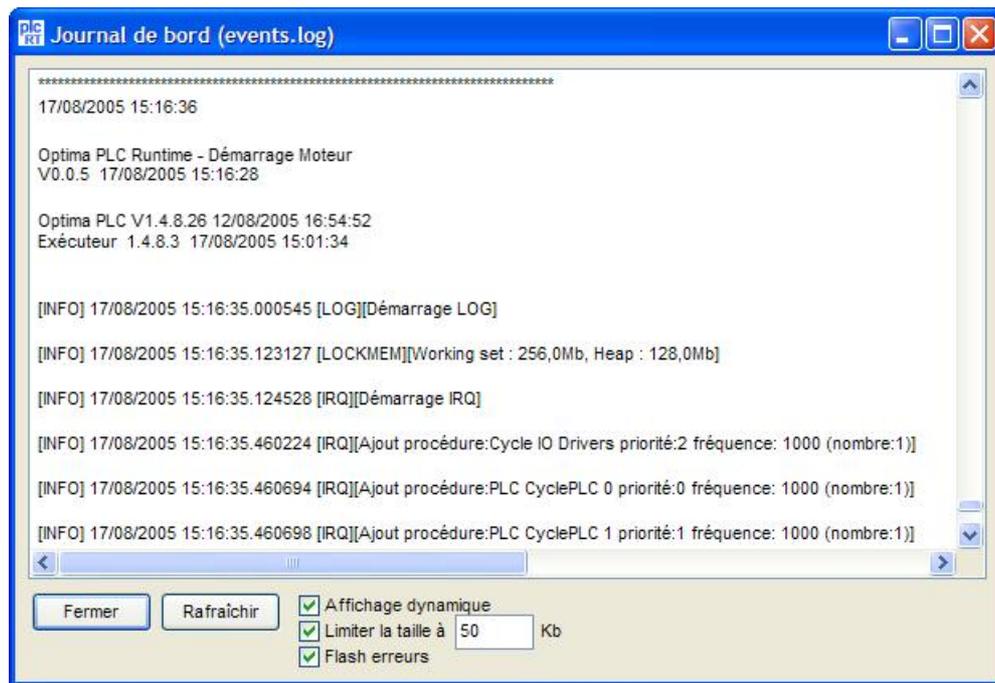
As far as possible, it is preferable that the total time consumed by Optima PLC remains relatively weak and don't pass 30% on average about the time available CPU. If this percentage is passed, some functions of Windows risk to be disrupted, in particular those making an important use of the interruptions, as the functions networks for example.

To reduce the percentage of occupation of the CPU by the runtime Optima PLC, a simple means consist in increasing the period of execution of the tasks of the application that doesn't require a period as weak that the

value by default of 1 millisecond.

4.3.5.3 Option Log

This option displays the visualization window of the runtime Optima PLC log.



The log receives the messages of information or error of the system, but also of the messages defined by the programmer.

These messages are stored in the Events.log file situated in the directory of OptimaPLC\log.

The display is done dynamically ; in the case where many messages are placed in the log, uncheck "Dynamic display" to freeze the content of the window.

The box "Limit the size to", associated to the input box situated on the right, permits to limit the size of the displayed data.

Finally, the box "Flash errors" permits to activate the blinking of the runtime window in the event an error message is placed in the log, until the consultation of this one.

Position of origin of an error

When a condition of error occurs, this one is signalled in the log. In the following example, a division by 0 :

```

[ERROR] 9/05/2006 18:02:24.710669 [IRQ][Divide Error (Code 200, @ 96D4DA4)]
 96D4DA4 Fichier:C:\FVCS\OptimaPLC\TEMP\_ST_PROG2.ST Proc:_FB_PROG2.EXECUTE Ligne:114
Colonne:6
 10098C41 Fichier:c:\fvcs\OptimaPLC\fpc\plc\st_procs.aps Proc:TFUNCTIONBLOCK__DOEXECUTE
Ligne:1359 Colonne:1
 96CD023 Fichier:C:\FVCS\OptimaPLC\TEMP\_ST_TASK1.ST Proc:_FB_TASK1.EXECUTE Ligne:29
Colonne:1
 10099B98 Fichier:c:\fvcs\OptimaPLC\fpc\plc\st_procs.aps Proc:TTASK__DOEXECUTETASK Ligne:1705
Colonne:1
 1009B784 (non trouvé)
 100B393B Fichier:c:\fvcs\OptimaPLC\fpc\lib\collec.aps Proc:CALLPOINTERLOCAL Ligne:134 Colonne:1
 100B3D6A Fichier:c:\fvcs\OptimaPLC\fpc\lib\collec.aps Proc:TCOLLECTION__FOREACH Ligne:296
Colonne:1
 1009B8B8 Fichier:c:\fvcs\OptimaPLC\fpc\plc\plc.aps Proc:CYCLEPLC Ligne:190 Colonne:1

```

The first line indicates the nature of the error, its date and its position in the space memory. The following lines correspond to the content of the call stack at the time of the error, and permit to localize this one.

The mentioned files are important: with the .ST extension, the files correspond to POU's of the application. A

double click on the corresponding line provokes the display of the POU in the workshop if this one is in function at this moment.

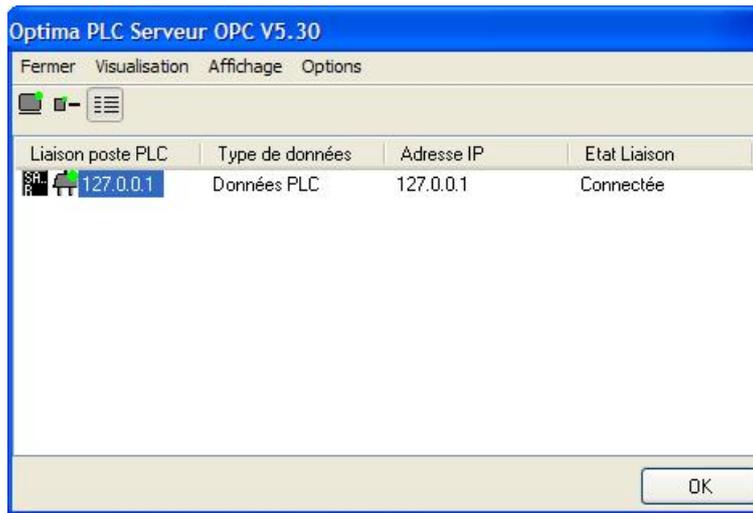
The files with the .APS extension correspond to the internal modules of Optima PLC.

Above, in the example a double click on the first line provokes (concerning the _ST_PROG2.ST file) the display of the POU Prog2 back in the workshop, the cursor placed to the site of the code having provoked the division by 0.

4.3.5.4 Option OPC server

This option permits the display of the current information concerning the OPC server integrated to Optima PLC.

These information are only available if the OPC server is working, what is the case if at least one client tempts to connect to it. The client the most frequently connected is the supervisory integrated to Optima PLC, but it can also be about an outside client application.



The displayed window permits to know the present state of the connection. The IP address of the connected PLC is mentioned (here 127.0.0.1 correspond to the local address, as definite by "Local_Host").

4.3.5.5 Option Lists of sockets

This option provokes the display of a diagnosis window listing all sockets (networks links) managed by the application.

No	Port	Connecté	Etat	Adresse distante	Adresse locale	Type	Statut
1	10203		Listening			TCP	Aucune erreur(0)
2	502	X	---	192.189.207.59	0.0.0.0	UDP	Aucune erreur(0)
3	10255	X	Established	127.0.0.1	127.0.0.1	TCP	Aucune erreur(0)
4	10247		Listening			TCP	Aucune erreur(0)
5	10247		Listening			TCP	Aucune erreur(0)
6	10247		Listening			TCP	Aucune erreur(0)
7	10247		Listening			TCP	Aucune erreur(0)
8	10247		Listening			TCP	Aucune erreur(0)
9	10247		Listening			TCP	Aucune erreur(0)
10	10247		Listening			TCP	Aucune erreur(0)
11	10247		Listening			TCP	Aucune erreur(0)
12	10247		Listening			TCP	Aucune erreur(0)
13	10247		Listening			TCP	Aucune erreur(0)
14	10247		Listening			TCP	Aucune erreur(0)
15	10247		Listening			TCP	Aucune erreur(0)


```

Adresse IP distante      :
Adresse IP locale       :
No de port distant      : 0
No de port local        : 10203

Type de connexion       : TCP serveur (Serveur disponible)
Etat actuel de la connexion : Listening

Octets reçus            : 0
Octets envoyés          : 0

```

Fermer

For every socket are mentioned various useful information: its connected state or no, the distant address, the numbers of port, etc...

This list contains the sockets based on TCP (links with protocol of connection establishment) and on UDP (simple links by telegram, no protocol of connection establishment).

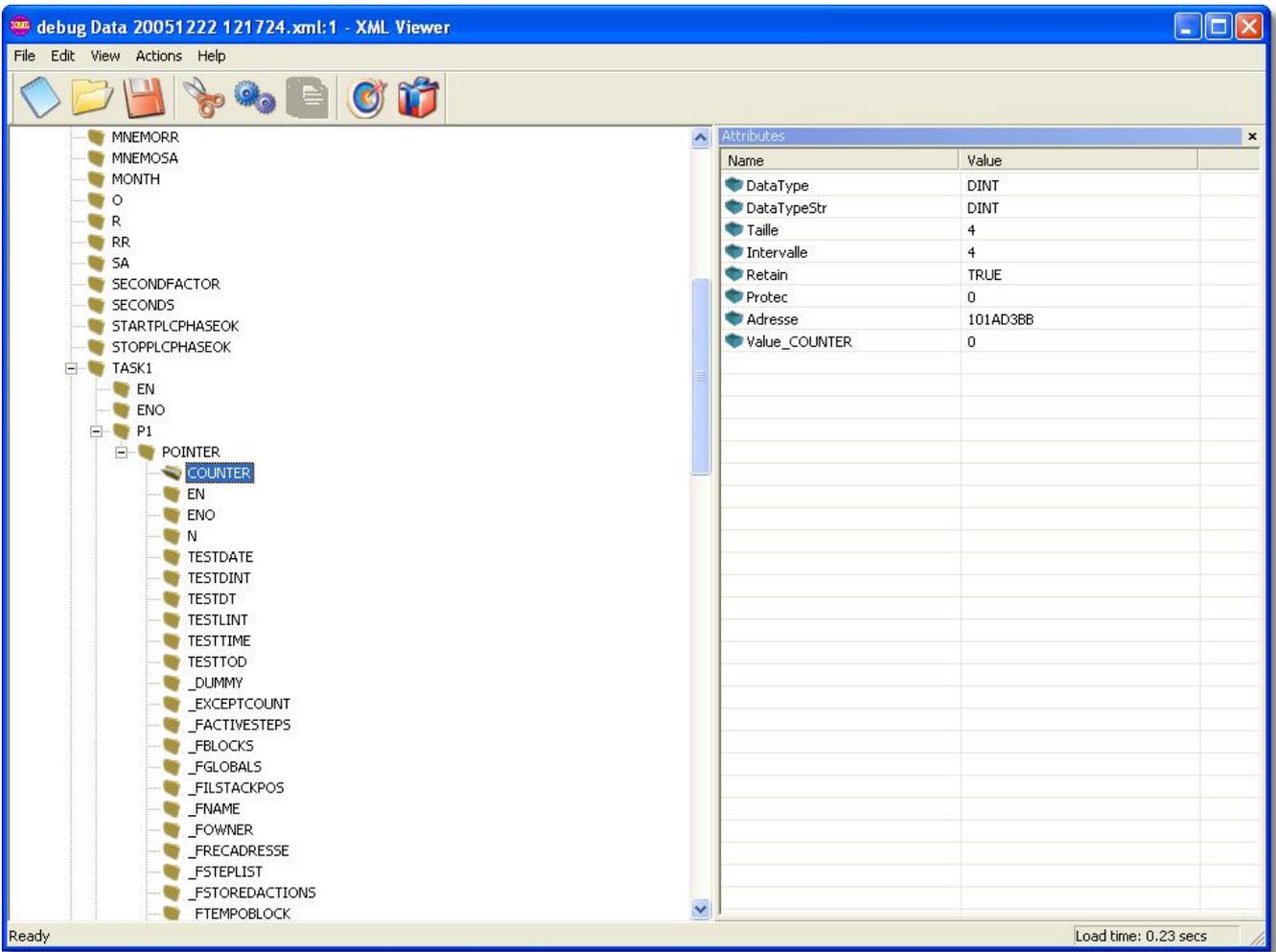
4.3.5.6 Option to "Generate a registration of diagnosis"

This option provokes the save of the complete PLC data in a goal of diagnosis or debug.

The name of the file is "Debug Data" follow-up of the date and the hour. The extension is .XML.

This file is also generated in output of application if the option "Debug information" has been selected for the compilation.

The generated file can be consulted in a program of XML visualization, as for example the free software "XML Viewer" of MindFusion (<http://www.mindfusion.org>).



The file only contains the data whose value is non nul, this in order to limit the size of the file and its time of loading.

4.3.6 Menu Help

Option Help	display this file of help.
Option About	Display the version form of the application and the runtime Optima PLC. <div data-bbox="756 1384 1278 2027" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p>A propos de ...</p> <p>Application Optima PLC Runtime</p> <p>V0.0.26 02/08/2006 17:42:50</p> <p>Optima PLC V2.1.0.1 02/08/2006 17:27:22 Exécuteur 2.1 02/08/2006 17:38:04</p> <p>OPTIMALOG 18 rue Charles Péguy F-37540 Saint Cyr sur Loire FRANCE</p> <p>email : contact@optimalog.com</p> <p style="font-size: 2em; font-weight: bold; color: #00AEEF;">OptimaPLC</p> <p style="text-align: right;"> <input type="button" value="Versions"/> <input type="button" value="Fermer"/> </p> </div>

4.4 Possible problems to the starting of the runtime

4.4.1 No "retain" data file at the starting of the application

At the starting, the application loads a back-up file of the retain variables. if no file is available, the following message is displayed :



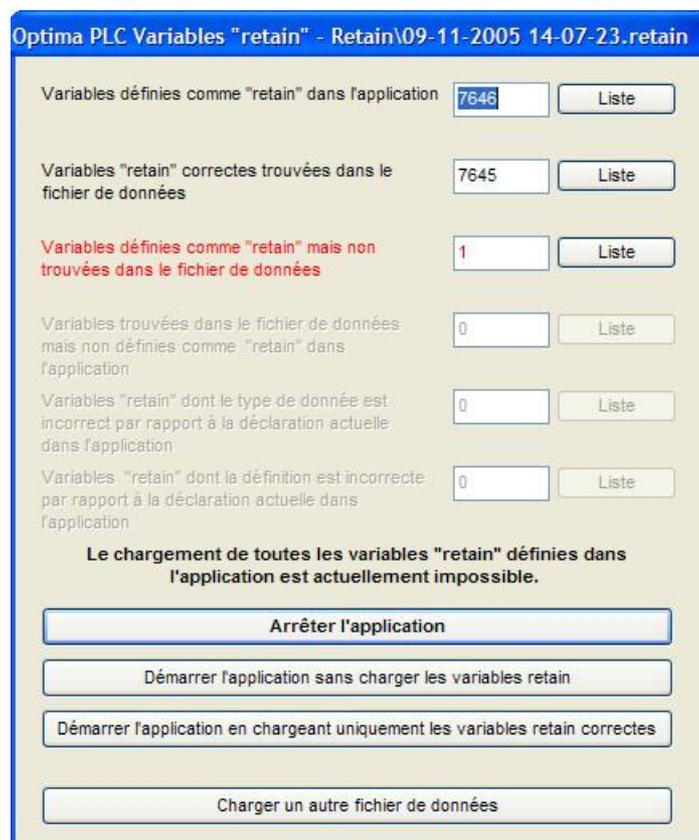
As the message indicates it, it is a normal situation at the time of the first starting of the application. The retain files are saved then cyclically and to the output of the application, the absence of such a file is then an anomaly.

By clicking on Yes, the variable retain is initialized by default with their value and the application is executed. By clicking on No, the application is stopped.

The retain files are stored in the subdirectory "retain" of the project.

4.4.2 Modification of the structure of the "retain" data

If a change intervened in the declaration of the retain variables since the last starting of the application, this one is detected by the runtime. A window as the one that follows is displayed then :

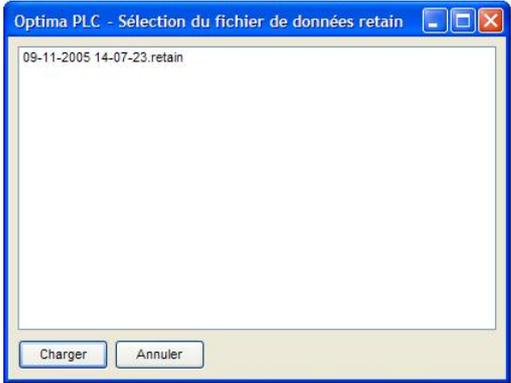


In this example, a variable retain named "TestRetain" has been added. It has not been saved therefore in the previous retain file. By clicking on the button "list" situated in front of the wording in red, the following window is displayed :



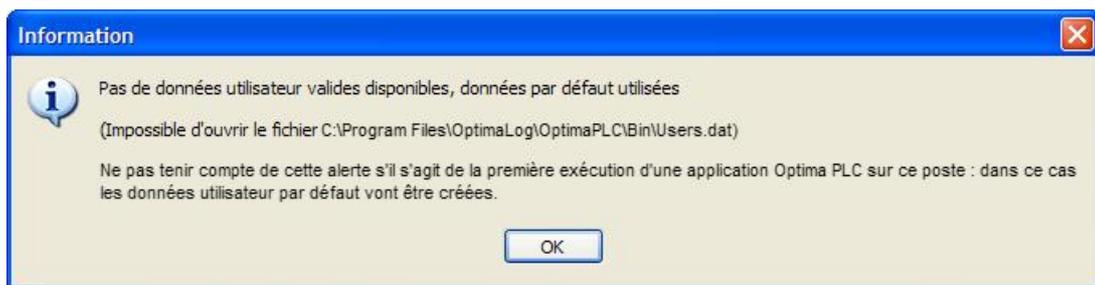
The set of the variables concerned by an anomaly is mentioned in the list.

Besides the consultation of the lists of variables, the options possible of the first window are the following :

Stop the application	The application is immediately stopped.
Start without loading the retain variables	The file of retain variables is ignored, the variables are all initialized to their default value.
Start loading the correct retain variables only	The retain variables for which the value stored in the file is correct are initialized with this one, the other are initialized to their default value.
Load another file of data	<p>This option permits to choose another retain file among those present.</p>  <p>The list of the available files is displayed. To choose one, place the cursor over then click to load it.</p> <p>The file is then analyzed, and the possible anomalies are signalled in the same way as previously.</p> <p>Attention: the consequences of the loading of a retain file non up to date can be important according to the application.</p>

4.4.3 Absence of data concerning the users

At the starting, the runtime loads a file of data concerning the users (program files\OptimaLog\OptimaPLC\bin\users.dat). If this file is not found, the following message is displayed :



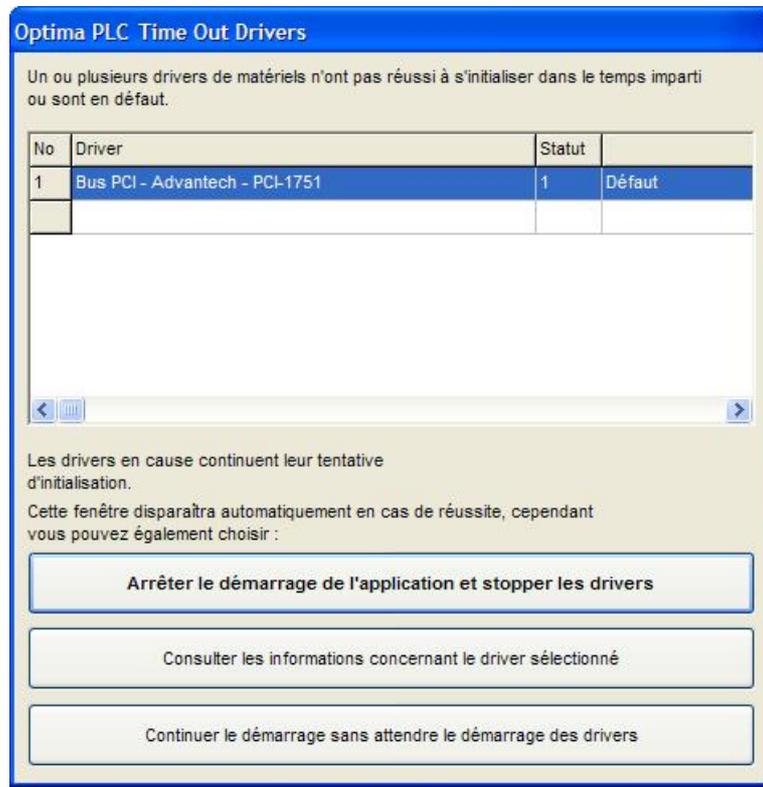
This situation is normal at the time of the first launching of an application Optima PLC on the PC, in this case ignore it.

If it is not the first launching of an application, the users.dat file is either damaged or inaccessible.

After having clicked on OK, user's data are restored by default, and a new users.dat file is created.

4.4.4 Problem of starting of a hardware driver

At the starting of the application, the runtime starts with initializing the drivers. At the end of 10 seconds, if one or several drivers didn't finish their initialization, the window below is displayed :



The list of the drivers posing problem is displayed. However the continuous initialization attempt. If this one ends, the driver disappears of the list. If it was the only one of the list, the initialization of the application continues normally.

During the time of display of this window, the application is in phase "[Start drivers timeout](#)^[320]". In this phase, no PLC task is executed.

The user has three options :

Stop the starting	The already initialized drivers are stopped, then the application is stopped too.
Consult the information concerning the selected driver	The window of diagnosis concerning the driver is displayed. The information that it contains depend on every driver.
Continue the starting	After a confirmation, the application continues its starting and pass in mode "Start PLC" then "Fault."

5 Management of the users

5.1 Presentation

After the phase of conception, when the application and the development tools are installed on the final hardware, different intervening parties can be brought to use the available functions. It is then necessary to control the accesses to the software in order to filter the authorizations according to the user.

Every software integrated in Optima PLC has a common advanced management of the users.

This functionality can be activated or can be deactivated separately in every software. When it is activated, it becomes necessary to identify before doing an action in the software.

A list of users common to the different software is held up to date. Every registered user has an identifier (or "login"), a group of adherence that defines its rights, and a password. At the time of the connection, the user must hit his identifier and his password. According to his group, he has then access or no to the different functions of the software.

At the end of a delay of five minutes without action of the user, the connection is cut and the user must identify again. The connection in one of the Optimalog software makes an immediate automatic connection in all others (if they use the management of the users) : in this manner it is not necessary to identify every time that one passes from a software to the other.

The groups of users are defined of the following manner :

Group	Prerogatives
Admins	Group permitting the whole set of possible and authorized operations according to the license installed. The author and installer of an application must belong to this group to program and to define the configuration.
Maintenance	Group authorized to most operations of edition and configuration, except the programming.
Supervisors	Group authorized to visualize and to edit the working parameters of an application.
Operators	Group whose accesses are limited to the visualizations of the parameters and to the actions foreseen by the author.

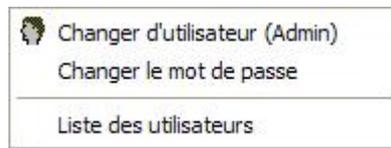
Attention

The Admin user (of the Admins group) is created by default with the "Admin" password. When the management of the users is deactivated in a software, this user is taken by default.

It is **greatly advisable** to modify by default the access password of the Admin user.

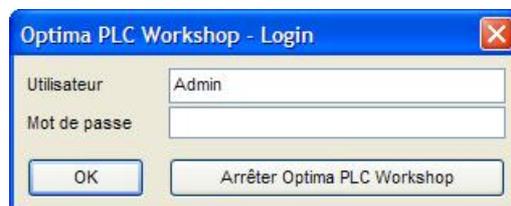
5.2 Menu Users

This menu regroups the options permitting to manage the list of the authorized users. It only appears if the mode "users management" has been activated in the options (these options can vary from a software to the other).



5.2.1 Option Change user

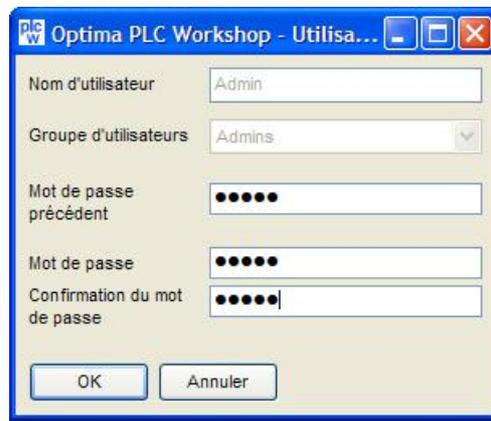
This option has for immediate effect to disconnect the current user, and provokes the display of the login window.



It is no more possible to use the software without hitting an identifier and a correct password.

5.2.2 Option to Change the password

This option allows the current user to modify his access password. By security, the previous password must be re-entered, and the new must be entered two times.



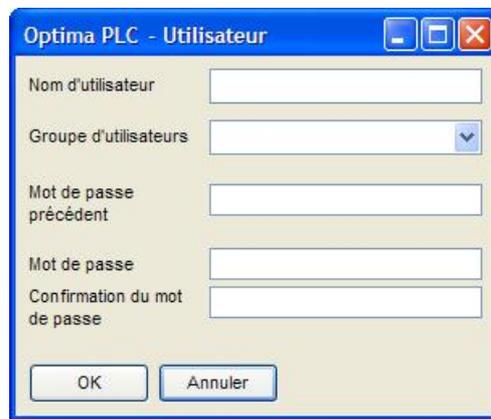
5.2.3 Option Lists some users

This option provokes the display of the users list.



If the current user has sufficient rights, it is possible to create (New button), to modify or to suppress users of the list. However, a user can only act on the accounts that are subordinate to him (unique exception: an admin can create another admin, but is then not able to modify it).

While clicking on New, the window of the user's information is displayed :



All fields are initially empty. User's name can only be entered at the creation, it cannot be modified thereafter.

6 Programming

Programming with Optima PLC takes the principles defined in the norm IEC 61131-3. Some variables types as well as a certain number of functions have been added in order to offer supplementary easiness.

Broadly speaking, a program is composed of :

- declarations of types

- declarations of variables
- instructions (simple instructions, calls of functions, calls of POU...)
- commentaries (framed by " * "and "*" according to the norm, Optima PLC also accepts "/" in beginning of commentary)

6.1 Data types

6.1.1 Integer

The integer types defined in the norm are :

Type	Description	Size in bits	Range	
SINT	Integer short	8	-128	127
INT	Integer	16	-32768	32767
DINT	Integer double	32	-2^{31}	$2^{31}-1$
LINT	Integer long	64	-2^{63}	$2^{63}-1$
USINT	Integer short non signed	8	0	255
UINT	Integer non signed	16	0	65535
UDINT	Integer double non signed	32	0	$2^{32}-1$
ULINT	Integer long non signed	64	0	$2^{64}-1$

Unusually, a variable can be defined of ANY_INT type, for example in the inputs of a function. The variable is then compatible with all integer types. It is stored like a variable of LINT type.

The integer values can be written in a decimal format (example: +65, -57, 0...) or in one of the following bases :

- binary: the values are preceded then of 2# (example: 2#00110101)
- octal: the values are preceded then of 8# (example: 8#020)
- hexadecimal: the values are preceded then of 16# (example: 16#C2)

For more of readability, the _ character can possibly be inserted in the middle of the numbers (example: +1_000_000).

6.1.2 Real

The real types defined in the norm are :

Type	Description	Size in bits	Beach of values
REAL	Real number	32	$\pm 1.5 \times 10^{45} .. 3.4 \times 10^{38}$
LREAL	Long real number	80 It is a difference in relation to the norm, that defines for this type a size of 64 bits	$\pm 3.6 \times 10^{4951} .. 1.1 \times 10^{4932}$

Unusually, a variable can be defined of ANY_REAL type. The variable is then compatible with all real types. It is stored as a variable of LREAL type.

The values can be written with one point to separate the integer value of the decimals and the symbol for the exponent is "e" or E (example: -45.68E7).

6.1.3 Duration

The type used for the durations is the TIME type.

The TIME values are normalized with the letters :

- d: days
- h: hours
- m: minutes
- s: seconds
- ms: milliseconds

They can be written with to 2 formats :

- short format : preceded of T#, the value is written without discontinuity between the numbers and the letters (example: T#3d45m2s)
- long format : preceded of TIME#, the value is written by separating the parts by the _ sign (example: TIME#2d_7h_10m)

The values of TIME type are stored on 64 bits with a resolution of the microsecond. The maximal duration is of about 300000 years.

6.1.4 Time stamp

The types date and time defined in the norm are :

Type	Description	Size in bits	Format
DATE	Date of the calendar	64	D#year-month-day or DATE#year-month-day
TIME_OF_DAY	Hour of the day	64	TOD#hour:minutes:seconds or TIME_OF_DAY#hour:minutes:seconds
DATE_AND_TIME	Date and hour	64	DT#year-month-day-hour:minutes:seconds or DATE_AND_TIME#...

The values are written by specifying the followed type of "#" then by separating the fields of the date by "-" and separating the fields of the hour by ":" (example: May 10, 2004 to 11h09 is written DT#2004 -05-10-11: 09:00).

All time stamp values are stored on 64 bits with a resolution of the microsecond.

6.1.5 Characters string

The type dedicated to the characters strings is the type "string."

The string is written between 2 signs ', and can contain the characters of control next one :

Character of control	Interpretation
\$\$	To use the \$ character in a string
\$'	To use the character ' in a string
\$L or \$l	Jump of line (line feed)
\$N or \$ n	New line

\$P or \$ p	Jump of page (form feed)
\$R or \$ r	Carriage Return
\$T or \$ t	Tabulation

The string type occupies 256 bytes and can contain a maximum of 255 characters. The first byte of the zone of data occupied is the "dynamic size" of the string, that is to say the number of meaningful characters.

A variable of string type also behaves like a table of CHAR, every element of the table, of 0 to 255, capable to be affected separately.

The string type can be drifted to produce strings of shorter length.

Example

the string 'Price software d\$'un: \$ L\$R1000\$\$'
 permits the following display: Price of a software :
 1000\$

6.1.6 Character

The character type corresponds to an ASCII value coded on 8 bits.

The characters can be specified with the same syntax that the strings of length 1 character.

6.1.7 Bits string

The "bit string" type defined in the norm are :

Type	Description	Size in bits	Values
BOOL	boolean	1 (binary format used 8 bits)	0 (or false) or 1 (or true)
BYTE	string of 8 bits	8 bits	
WORD	string of 16 bits	16 bits	
DWORD	string of 32 bits	32 bits	
LWORD	string of 64 bits	64 bits	

As for the integer, it is possible to write the binary information in decimal, binary (2#) or hexadecimal (16#) format.

For more readability, the _ character can be inserted in the middle of the numbers (example: 2#1000_0100).

6.1.8 Generic types

The generic types cannot be used to declare some variables. These types are mentioned here because they are used in the declaration of the standard blocks (example: the AND block can be used with all bits strings).

The data types are structured according to a hierarchy that permits to define the generic types that regroup similar types. These types are :

ANY			
	ANY_NUM		
		ANY_REAL	LREAL, REAL
		ANY_INT	LINT, DINT, INT, SINT ULINT, UDINT, UINT, USINT
	ANY_BIT		

	LWORD, DWORD, WORD, BYTE, BOOL
STRING	
ANY_DATE	
	DATE_AND_TIME, DATE, TIME_OF_DAY
TIME	

6.1.9 Derivative types

It is possible to create new data types derived of the preset types.

The declaration of a new type is framed by the words key TYPE and END_TYPE.

Example

```
TYPE
  TypeExemple: string;
END_TYPE
```

The types can be defined by the simple declaration of a variable or while using formats of data more structured as :

- [the data structures](#) ^[349]
- [the enumerated types](#) ^[349]
- [the ranges of value](#) ^[350]
- [the arrays](#) ^[350]

6.1.9.1 Data Structure

The data structure is a set of data of different types.

Its declaration is framed by the words key STRUCT and END_STRUCT.

Example

```
TYPE
  StructureExample :
    STRUCT
      Variable1OfStructureExample : DINT;
      Variable2OfStructureExample : DWORD;
      Variable3OfStructureExample : TIME;
    END_STRUCT
END_TYPE
```

If a variable STR is declared then with the StructureExample type, it will be possible to use the following expressions :

```
STR.Variable1OfStructureExample
STR.Variable2ofStructureExample
STR.Variable3ofStructureExample
```

6.1.9.2 Enumerated Types

It is possible to affect some names to values for a given type of variable.

The enumerated types permit a better readability of the program thus; on the other hand the control of type done by the compiler minimizes the risk of error, in relation to the use of numeric values.

In the declaration of the type, the names of the different states are mentioned in brackets and separated by a comma. For the use of a value, write the name of the variable followed of the sign # to specify the state then.

Example

```
TYPE
  ExampleEnumType : (STOP, RUN, FAIL);
END_TYPE
```

```
...
SystemState:= ExampleEnumType#STOP;
```

6.1.9.3 Ranges of values

It is possible to limit the possible values of a variable by defining a type with an interval of values. To define the interval, write in brackets, after the type, the limit values separated by 2 consecutive points.

Example

```
TYPE
    ExampleInterval : INT (-4..+22);
END_TYPE
```

6.1.9.4 Arrays

The keywords ARRAY and OF are used to define the tables of data. The limits of the table are defined between hooks. Every dimension is defined by an indication of beginning separated by 2 consecutive points of the end indication. The sign "," permits to separate the definition of 2 dimensions. The data type of the table elements is indicated after the OF keyword.

Example

```
TYPE
    ExampleTable2D : ARRAY [1..5 ,2.. 10] OF DWORD;
END_TYPE
```

6.2 Preset variables in Optima PLC

6.2.1 Tables of variables

Those are tables of different types values, permitting to simplify the programming and the data declaration. These variables can be consulted directly and can even be modified in the runtime in special pages.

In most cases, the size of the tables is defined by a modifiable value in the project options (System constants).

Name	Indication of beginning	Indication of end	Type	Use	Direct access
R	0	Defined in Constants System	DINT	Integer numeric registers	%MD0.x
RR	0	Defined in Constants System	LREAL	Real numeric registers	
SA	0	Defined in Constants System	BOOL	Flags modifiable in the tables by the operator	%MX0.x
L	0	Defined in Constants System	BOOL	General flags	%MX1.x
C	0	Defined in Constants System	STRING	Characters strings	%MS0.x <i>to see note</i>
DT	0	Defined in Constants System	TIME	Temporizations. The value of these variables, once loaded with a time value, is automatically decreased with the passing time until 0.	%MT0.x <i>to see note</i>
A	0	Defined in Constants System	BOOL	Flags of alarms (provoke the registration and the apparition of a message)	%MX2.x
I	0	Depends on the installed drivers	BOOL	Digital Inputs	%IX0.x
ANI	0	Depends on the installed drivers	LREAL	Analogical inputs	%IR0.x <i>to see note</i>
O	0	Depends on the installed drivers	BOOL	Digital outputs	%QX0.x

ANO	0	Depends on the installed drivers	LREAL	Analogical outputs	%QR0.x to see note
------------	---	----------------------------------	-------	--------------------	-----------------------

note : this syntax of direct access is specific to Optima PLC.

Mnemonic

The preset variables can be used directly in the program without any declaration. However, it is possible to associate them a mnemonic and a descriptive text. The mnemonic can be used in the blocks instead of the preset variable. The descriptive text is displayed in the windows of the runtime preset variables.

For example, here is the definition of a mnemonic and a descriptive for the variable R10 :

```
CounterBad AT R10: dint; // counter of bad pieces
```

The mnemonic CounterBad and the text descriptive "counter of bad pieces" is now (commentary situated at the end of declaration line) associated to the R10 register.

The following declaration, in total agreement with the IEC61131 norm, is also possible and give the same result :

```
CounterBad AT %MD0.10 : dint; // counter of bad pieces
```

To be taken correctly in account, this kind of declaration must be in a VAR..END_VAR clause of a global variables block. On the other hand, the type of variable specified must be imperatively the one of the preset variable.

Tables of the mnemonics and descriptive texts

The tables "MnemoXX" and "TextXX" of characters strings contain the mnemonics and descriptive texts of the preset variables subject to a definition of mnemonics.

Besides a variable "Text"mnemonic is also created.

In the case of the above stated example, the following variables are therefore available :

Variable	Contained
MnemoR [10]	"CounterBad"
TextR [10]	"counter of bad pieces"
TextCounterBad	"counter of bad pieces"

6.2.2 Particular cases

6.2.2.1 A Alarms

The A alarms are boolean variables predefined in a table. Their interest is to permit a simple management of alarm messages intended to the operator (another more elaborate management is proposed with the software *Optim'Alarm*).

When one of the alarms is put to 1 or 0, a line is added in the application logfile. Besides, when set to 1, an information window showing the active alarms can be displayed (see the options of the runtime).

To be exploited fully, the alarms must be used with declaration of mnemonic and comment :

Example of declaration

```
FaultAirPression A5 at: bool; // Defect of air pressure
```

In this example, set to 1 FaultAirPression will provoke the display of "Defect of air pressure" in the window of the active alarms. This message will also be mentioned in the logfile.

6.2.2.2 DT temporizations

The DT temporizations are variables of Time type predefined in a table. Their particularity is that their value, if it is not zero, is automatically decreased of the elapsed time until the value 0.

This possibility makes especially simple the programming of temporizations. Simply load a variable DT with time

wished, then test its passage to 0. It is also possible to test an intermediate value.

6.2.3 Various variables

Time stamp variable (all of DINT type) :

Day	day of the month (1 to 31)
Month	number of the month (1 to 12)
Year	year
Hours	hour of the day
Minutes	minutes in the current hour
Seconds	seconds in the current hour
DayOfYear	number of the day in the year
WeekNb	number of the week in the year

6.3 Declarations

6.3.1 Declarations of variables

The declaration of variables takes place between 2 keywords, the first defines the type of declaration, the second END_VAR encloses the declaration part.

The declaration is composed of the name of the variable followed of ":" then of the type of variable and finish by " ;"

The syntax used here is the one of the norm IEC 61131-3.

Example

```
VAR
    Variable1: DINT;
END_VAR
```

The variables can be :

- global: the variables are accessible since all POU's situated after the declaration. In Optima PLC, a specific POU exists for this type of variable that permits to declare globally without having to redefine the variable as external (as defined in the norm) everywhere it is used.
- local: the variables are only usable in the POU where they have been declared (tab "Declarations" in Optima PLC).

The variables are classified then according to their type of use :

VAR	Function, function block, program : the variables are local (visible only in the POU) and are stored between 2 calls of the POU (static variables).
	POU of variables declaration : the variables are globally accessible to all POU's declared after.
VAR_TEMP	the variables are temporary, their value is not stored between 2 calls of the POU.
VAR_INPUT	the input variables are the parameters of the POU, they must be affected before or at the time of the call of the POU, but cannot be modified in the POU
VAR_OUTPUT	the output variables are the results of execution of the POU, they can only be affected by the POU
VAR_IN_OUT	the output variables are the parameters that can be modified in the POU.
VAR_GLOBAL	the variables are global (authorized only in a block of program). This syntax is compliant to the norm IEC 61131-3, but global variables declared in POU of variables declarations are more convenient.
VAR_EXTERNAL	the variables are local to the POU but are bound to global variables defined in a block of program with VAR_GLOBAL.

Direct access in memory	<p>the name of the variable begins with % and is followed of 2 letters whose interpretation is for the first the type of zone memory, for the second the length of the data (to see table below). A value of indication is added then to these 3 characters.</p> <p>Example : %MB100: the variable is the 100th byte in internal memory</p>
--------------------------------	--

First letter	Interpretation	Second letter	Interpretation
I	Site memory in input	X	Bit
Q	Site memory in output	B	Byte (8 bits)
M	Internal memory	W	Word (16 bits)
		D	Double word (32 bits)
		L	Long word (64 bits)

Coding for direct access in memory

Case of the functions

In the case of a function, the possibilities of declaration are reduced. In particular, the VAR declaration (static variables) is not allowed.

6.3.2 Attributes of variables

The variables can have additional attributes defined by the following keywords :

- RETAIN: indicate that the value of the variables must be kept when there is disappearance of the power of the PLC (permits a safeguard in case of current cut)

Example

```
VAR RETAIN
Statistic1 : LINT;
END_VAR
```

- CONSTANT: the variable has a constant value during the execution of the program, this value is specified after the type with the sign " := "

Example

```
VAR CONSTANT
NumberOfPumps : SINT := 4;
END_VAR
```

- AT: to give a name to a variable specifying its memory site ; in Optima PLC it is thus possible to use this keyword to affect a mnemonic to a preset variable.

Example

```
VAR
MachineNumber AT %MB1;
MachineStop AT SA0;
END_VAR
```

6.3.3 POU instances

To be able to use a function block from another POU, it is necessary to create an instance of it in the declaration part. This declaration is quite similar the declaration of any variable.

Attention, the block instance must not be defined in the temporary variables (VAR_TEMP).

Example : the Machine1 program uses an instance of the SetPower function block

```
VAR
  SetPowerMachine1 : SetPower;
END_VAR
```

Remark : the fact to declare an instance of POU doesn't mean that the instructions of this POU will be executed; to execute this part of program, the call to "SetPowerMachine1" instance must be part of the program instructions.

6.4 Constant values

6.4.1 Numeric constants

Description	Examples
Integer literals	-12 0 123_456 +986
Real literals	-12.0 0.0 0.4560 3.14159_26
Real literals with exponents	-1.34E-12 or -1.34e-12 1.0E+6 or 1.0e+6 1.234E6 or 1.234e6
literals in basis 2	2#1111_1111 (255 decimal) 2#1110_0000 (240 decimal)
Labels in basis 8	8#377 (255 decimal) 8#340 (240 decimal)
literals in basis 16	16#FF or 16#ff (255 decimal) 16#E0 or 16#e0 (240 decimal)
Boolean "zero" and "one"	0 1
Boolean TRUE or FALSE	FALSE TRUE

Important : the integer values in basis 10 are compatible with the numeric types (SINT, USINT, INT,...), whereas the integer values in basis 2, 8 or 16 are considered like bits strings. To go beyond this restriction, use typed constants (see farther) :

INT#16#12 integer value coded in hexadecimal.
WORD#100 bits string coded in decimal

6.4.2 Constants characters strings

A characters string is a sequence of 0 to n characters, beginning and ending with a single quote (').

The sequence of the sign \$ followed of two hexadecimal numbers corresponds to the hexadecimal representation of a single character.

Example	Explanation
"	Empty string
'A'	string of length 1 containing the character A
' '	string of length 1 containing the character "space"
'\$'	string of length 1 containing the single quote character
'\$R\$L' '\$OD\$0A'	strings of length 2 containing the CR and LF characters.
'\$\$1.00'	string of length 5 correspondent to \$1.00

The sequences constituted of the character \$ followed of a special character are described below.

Combination	Interpretation
\$\$	Character Dollar
\$'	Single quote character
\$L or \$l	LF (line feed)
\$N or \$n	New line
\$P or \$p	FF (Form feed)
\$R or \$r	CR (carriage return)
\$T or \$t	Tabulation

6.4.3 Constants of duration

Description	Examples
Duration literals without underlining Short prefix	T#14ms T#-14ms T#14 .7s T#14 .7m T#14 .7h t#14.7d t#25h15m t#5d14h12m18s3.5ms
Long prefix	TIME#14ms TIME#-14ms time#14.7s
Duration literals with underlining Short prefix	<u>t#25h_15m</u> <u>t#5d_14h_12m_18s_3.5ms</u>
Long prefix	TIME# <u>25h_15m</u> time# <u>5d_14h_12m_18s_3.5ms</u>

6.4.4 Constants of date

Prefixes for the literals :

Description	Prefix
Date literals Long prefix	DATE#
Date literals Short prefix	D#
Time of day literals Long prefix	TIME_OF_DAY#
Time of day literals Short prefix	TOD#
Date and time literals Long prefix	DATE_AND_TIME#
Date and time literals Short prefix	DT#

Examples

Long prefix	Short prefix
DATE#1984-06-25 date#1984 -06-25	D#1984-06-25 d#1984 -06-25
TIME_OF_DAY#15:36:55.36 time_of_day#15:36:55.36	TOD#15:36:55 .36 tod#15:36:55 .36
DATE_AND_TIME#1984 -06-25-15: 36:55.36 date_and_time#1984 -06-25-15: 36:55.36	DT#1984 -06-25-15: 36:55.36 dt#1984 -06-25-15: 36:55.36

6.4.5 Typed constant values

The use of constant values sometimes presents a difficulty concerning their type, this one not being explicit.

For example, the constant value 10 is compatible with the following types: SINT, USINT, UINT, INT, UDINT,

DINT, LINT, ULINT. The constant value 1 is also compatible with the BOOL type.

These numeric values don't have a precise type therefore at the time of their assessment, the compiler allowing this uncertainty as a long time as possible.

It is however possible to force the type of all constant value while prefixing the constant with the type wanted follow-up of #, in the same way as for the TIME values, DATE, etc.

Examples

INT#15	integer value 15
INT#16#12	integer value 18
BYTE#10	bits string 2#1010
BOOL#1	value boolean true

6.5 Initialization of variables

The variables can be initialized of 2 ways :

- at the time of the declaration of the variable type

The value specified here corresponds to the value initialized at start of the application if the variable is not "[RETAIN](#)". If the variable is RETAIN, the value is used for the initialization only when no data retain concerning it is available.

- In a task executed in START phase, permitting the initialization before the passage in RUN.

6.5.1 Initialization of types and simple variables

A type or a variable can be initialized at the time of its declaration.

The value is affected then with the sign ":", and is placed at the end of the declaration.

Example

```
TYPE
  VarInitialized :      DINT   := 22;
END_TYPE
```

Special case: initialization of the tables

To initialize a table, the values must be separate by commas; when several consecutive elements receive the same value, it is possible to specify the number of elements to initialize and to indicate the value to affect in brackets.

Example : the table is initialized with the values 1 for the 1st element, 0 for the 3 next one and 10 for the last element

```
VAR
  TableStatistics: ARRAY [0..5]:= 1, 3 (0), 10;
END_VAR
```

6.5.2 Initialization of types and multi-elements variable

It is possible to initialize variables of type multi-elements.

Initialization of the tables

To initialize a table, the values must be separate by commas; when several consecutive elements receive the same value, it is possible to specify the number of elements to initialize and to indicate the value to affect in brackets.

Example : the table is initialized with the values 1 for the 1st element, 0 for the 3 next one and 10 for the last element

```
VAR
  TableStatistics : ARRAY [0..5]:= 1, 3 (0), 10;
```

```
END_VAR
```

All elements must not be initialized inevitably.

Initialization of structure

Every field of the structure can be initialized. At the time of the declaration, after having mentioned the type of the variable, the different fields are mentioned in brackets with the values affected to each of them. All fields must not be initialized inevitably.

Example

```

TYPE Limits
  STRUCT
    Minimum: DINT;
    Maximum: DINT;
    Average: DINT;
  END_STRUCT
END_TYPE

VAR
  LimitesTension      : Limits (Minimum:= 0, Maximum:= 10);
END_VAR

```

6.6 Programming in structured text language (ST)

The ST language is an evolved structured language, close to the Pascal language, permitting to define treatments and complex algorithms.

6.6.1 Expressions

An expression is a syntactic construction that can be evaluated, in order to provide a corresponding result of one of the types of data defined.

The expressions are constituted of operators and operands.

The operators are summarized below in the table.

	Operation	Symbol	Priority
1	brackets	(expression)	Maximal
2	Evaluation of function	identifier (argument list) example : LN(A), MAX(X, Y), etc.	
3	Exponentiation	**	
4	Negation	-	
5	Complement	NOT	
6	Multiplication	*	
7	Division	/	
8	Modulo	MOD	
9	Addition	+	
10	Subtraction	-	
11	Comparison	< , > , <= , >=	
12	Equality	=	
13	Inequality	<>	
14	AND Boolean	&	
15	AND Boolean	AND	
16	Exclusive OR Boolean	XOR	

17	OR boolean	OR	Minimal
----	------------	----	---------

The evaluation of an expression consists in applying the operators to the operands in the order defined by the priority.

For example, if A, B, C and D are of the Int type and have the values 1, 2, 3 and 4, then

B-C*ABS(D)	takes value 9
(A+B-C)*ABS(D)	takes value 0

6.6.2 Instructions

Summarized of the types of instruction.

	Instructions	Examples
1	Affectation	A := B; CV := CV+1; C := SIN(X);
2	Call of a function block	CMD_TMR (IN:=%IX5, PT:=T#300ms); A := CMD_TMR.Q;
3	RETURN Come out of a function or a function block	RETURN;
4	IF	D := B*B - 4*A*C; IF D < 0.0 THEN NROOTS := 0; ELSIF D = 0.0 THEN NROOTS := 1; X1 := - B/(2.0*A); ELSE NROOTS := 2; X1 := (- B + SQRT(D))/ (2.0*A); X2 := (- B - SQRT(D))/ (2.0*A); END_IF;
5	CASE Selection among several values	TW := BCD_TO_INT (THUMBWHEEL); TW_ERROR := 0; CASE TW OF 1,5 : DISPLAY := OVEN_TEMP; 2 : DISPLAY := MOTOR_SPEED; 3 : DISPLAY := GROSS - TARE; 4,6..10: DISPLAY := STATUS(TW - 4); ELSE DISPLAY := 0; TW_ERROR := 1; END_CASE; QW100 := INT_TO_BCD(DISPLAY);
6	FOR Repetition of a loop n time.	J := 101; FOR I := 1 TO 100 BY 2 DO IF WORDS[I] = 'KEY' THEN J := I; EXIT; END_IF; END_FOR;
7	WHILE Repetition while the condition is true.	J := 1; WHILE J <= 100 & WORDS[J] <> 'KEY' DO J := J+2;

		END_WHILE;
8	REPEAT Repetition until the condition is true.	J := -1; REPEAT J := J+2; UNTIL J = 101 OR WORDS[J] = 'KEY' END_REPEAT;
9	EXIT Permits to come out of an iterative loop before the terminal condition is satisfied.	EXIT;
10	Empty instruction	;

6.6.2.1 Affectation

Affectation permits to replace the value of a variable by the result of the evaluation of an expression.

The left part of an affectation is a variable, the right part is an expression of which the type of the result corresponds to the one of the variable.

Example

```
A: = B + C;
```

6.6.2.2 Call of function

The call of a function is done like part of an expression.

Example

The "Func" function has two integer parameters, and returns a real.

```
X: = Func (A, B) * 1.5;
```

6.7 Programming in instructions list (IL)

An IL program is constituted of a list of elementary instructions, near the assembly language.

6.7.1 Operators, modifiers, operands

Every instruction must begin on a new line. It is constituted of an operator, possibly of one or several modifiers, and of one operand.

An instruction can be preceded of a "label" followed by ":", and then a commentary.

Label	Operator (possibly followed of one or several modifiers)	Operand	Commentary
START:	LD	I100	(* PUSH BUTTON *)
	ANDN	SA100	(* NOT INHIBITED *)
	ST.	O80	(* FAN ONE *)

Remark : when a label is defined, ther must be no space between the label and the symbol ":".

The instructions manipulate the "accumulator", or "current result". It is a variable capable to take the different types of data. This result can be set temporarily aside during the evaluation of another result, what drives to the classic notion of stack.

The general working of an IL instruction, named OP for example, is the next one :

$$\text{Result:} = \text{Result OP operand}$$

For example, the instruction :

AND SA10

must be interpreted like :

Result: = Result AND SA10

	Operator	Modifiers	Operand	Significance
1	LD	N		Give to the current result the type and the value of the operand
2	ST	N		Store the value of the current result in the operand. The current result and the operand must have the same type.
3	S	Note 3	BOOL	If the current result is 1, force the operand to 1.
	R	Note 3	BOOL	If the current result is 1, force the operand to 0.
4	AND	N, (BOOL	AND boolean
5	&	N, (BOOL	AND boolean
6	OR	N, (BOOL	OR boolean
7	XOR	N, (BOOL	Exclusive OR boolean
8	ADD	(Addition
9	SUB	(Subtraction
10	MUL	(Multiplication
11	DIV	(Division
12	GT	(Comparison >
13	GE	(Comparison >=
14	EQ	(Comparison =
15		(Comparison < >
16	THE	(Comparison <=
17	LT	(Comparison <
18	JMP	C, N	Label	Jump to the instruction that follows the label
19	CAL	C, N	Name	Call of a function block
20	RET	C, N		End of execution of the block
21)			Evaluation of a deferred operation

A modifier is an element that comes to modify the operator, it stands at the end of the operator (do not put any space between the two). Example: ORN.

Modifier N provokes a negation of the operand.

Modifier C indicates that the instruction must not be done that if the current result is equal to 1.

Example :

JMPC Label1

This instruction provokes the jump to the instruction that follows Label1 only if the current result is equal to 1.

Finally, the modifier (permits to defer the operation. For example :

```
LD R10
ADD (R11
MUL 10
)
```

Must be interpreted like :

```
Result: = R10
Result: = Result + (R11 * 10)
```

The modifiers (can be overlapped several times.

The operator) must appear as many times as the modifier (.

Example :

```
LD R5
ADD (R10
MUL R11
DIV (R12
ADD R13
)
)
```

6.7.2 Call of functions

The functions must be called by writing the name of the function in the operator field. The result of the function is placed in the current result.

The functions can be called in a formal or not formal way.

Formal call :

The name of the function is followed of the modifier "(", then the affectations of the input parameters (an affectation by line, written to the ST format without ";" at the end), the end of affectations is signalled by the operator ")".

Example :

```
Call a function CALCULATION including 2 input parameters, Value1 and Value2.
The formal call of this function is the next one :
```

```
CALCULATION(
Value1: = 10
Value2: = 5
)
```

Not formal call :

The current result is used like first parameter of the function. The name of the function is possibly followed of a second parameter.

Examples :

```
LD 30.2
CO (* calculation of the cosine in the current result *)
LD 10
CALCULATION 5 (* equivalent to the example describes in the part formal Call *)
```

6.7.3 Call of function blocks

A call of function block can be made under different ways, as describes in the following table.

The following declaration is supposed made in these examples :

```
VAR
  C10: CTU;
END_VAR
```

	Description	Instructions
1	CAL with list of inputs	CAL C10 (CU:=%IX10, PV:=15)
2	CAL with LD instruction and ST for the inputs	LD 15 ST C10.PV LD %IX10 ST C10.CU CAL C10
3	Use of input operators (see following table)	LD 15 PV C10 LD %IX10 CU C10

Input operators for the standard function blocks :

	Operators	Type of function block
4	S1, R	SR
5	S, R1	RS
6	CLK	R_TRIG
7	CLK	F_TRIG
8	CU, R, PV	CTU
9	CD, LD, PV	CTD
10	CU, CD, R, LD, PV	CTUD
11	IN, PT	TP
12	IN, PT	TON
13	IN, PT	TOF

These operators do the loading of the block input with the same name, with the value of the current result, that must be of the same type. The block is called then.

Example :

```
LD 100000
PV CTD1      (* Loading of the down counter with the value 100000 *)
LD 1
CD CTD1      (* trigger of the down counter *)
```

6.8 Programming in Apigraf language

The Apigraf language is a language dedicated to the PLC programming, based on simple textual boolean conditions.

6.8.1 Typical boolean instruction

An Apigraf boolean instruction can decompose itself in three columns :

Condition	Action	Commentary
-----------	--------	------------

The width of every writing zone doesn't have a particular importance. Only two rules must be respected: the conditions must begin at the beginning of line, contrary to the actions that must not begin at the beginning of line.

Example

```
!Start1 & \EmergencyStop      Motor1      ! Motor 1 starting
!ELSE                          \Motor1      ! Motor 1 stop
!                               ! (ELSE part is optional)
```

Explanation

- **Condition**

written boolean expression inevitably in beginning of line.

- **Action**

if the condition is true, the actions specified in this part are executed. The part "ACTION" can include several lines, every line containing only one action.

An ACTION must be preceded of at least a space or of a tabulation. Indeed, unlike the conditions, the actions must not begin at the beginning of line.

For the ACTIONS, one also speaks of PROCESS.

Attention: in the languages of the norm IEC 61131-3, the word "action" relates to a concept a little different from the one of Apigraf: it is here about any instruction doing something in answer to a condition.

- **Commentary**

They are always preceded of the sign "!".

In the previous example, the condition :

```
Start1 & \EmergencyStop
```

is the boolean expression conditioning the action to do. If this one is true (to 1), the "Motor1" output is activated, otherwise it is deactivated (instruction "\Motor1").

The part "ELSE" is optional in a condition. The word ELSE can be placed or no at the beginning of line.

Example

```
!Start1 & \EmergencyStop
|           Motor1
|           Lamp1
!Else      \Motor1      ! This part
|           \Lamp1      ! is optional
```

6.8.2 Condition (boolean expression)

A condition, in APIGRAF, is a boolean expression whose result is true (1) or false (0). This result comes from logical operations on boolean or numeric values.

The brackets "(" and ")" permit to frame the conditions, in order to get their correct evaluation.

The boolean operators are the next one :

&	AND logical
#	OR logical
\	Logical negation
XOR, < >	OR exclusive
=	equality of two boolean values

The conditions formed of boolean values combined by these operators are correct. significance logical of the signs = (equal) and < > (different)

- **Equality**

The two following writings are equivalent :

```
!(L1 = L2)           action1
!(L1 & L2) # (\L1 & \L2)  action1
```

- **Difference**

The two following writings are equivalent :

```
!(L1 <> L2)           action1
!(L1 & \L2) # (\L1 & L2)   action1
```

- **Examples of conditions**

```
!L1                ! true if L1 is true
!L1 <> A2           ! true if (L1 and not A2) or (not L1 and A2)
!L1 & \SA2 & (L3 = L4) ! true if L1 and not SA2 and
!                ! ((L3 and L4) or (not L3 and not L4))
```

Above, in the conditions the boolean values are boolean variables, as the L flags, the alarms A, the SA, or the I/Os : it is the simplest case.

- **Test on numeric values**

However, the boolean values can be also the result of comparison on numeric values :

```
!R10 > 500          action1
```

action1 will be activated if R10 is superior to 500

```
!L1 & (R10 > 1000) & (R1 + R2 - R3 = 0)
```

Here, the result of the comparison of R10 with 1000 gives a boolean value directly usable in the condition. Some intermediate numeric calculations can also be done, as here $R1 + R2 - R3 = 0$ the result of the calculation is compared to "0" (to see the numeric operators below).

The conditions must always begin at the beginning of line.

The conditions on numeric variables must have brackets if they are combined with other conditions.

6.8.3 Example of conditions and actions

Examples of correct conditions

```
!DT6 = 5000        ! true if DT6 is equal to 5000
!R4 * 2 = R7 div 3 ! true if 2 times R4 is equal to R7 divided by 3
!Timer <= 25000    ! true if Timer is smaller or equal to 25000
!(I5 # (DT5 = 0) # \SA5) & (Timer < R6) & !I7
```

Note

The boolean expression defining a condition must begin at the beginning of line. A numeric comparison must be placed in brackets if it is combined to another by an operator :

```
!R5 > 500          ! correct
!R5 > 500 & SA2    ! incorrect
!(R5 > 500) & SA2  ! correct
```

Examples of correct actions

Note

The "process" is the set of the actions that will be activated or no according to the condition that precedes. An instruction of the process is always preceded of spaces or a tabulation, that indicate that it is about an action to do (to the difference of the conditions that begin at the beginning of line).

Actions on boolean variables

```
! O5                ! activate the output 5
! \HL3              ! disactivate the HL3 output
! \O5 ,8            ! disactivate the outputs 5 and 8
! O7 = SA4 & (I3 #I4)! O7 is true if
!                  ! the boolean equation on the right
```

```

|           ! of the sign is true
| L4       ! activate the logical flag
|           ! L4 to "true"
| \SA6     ! deactivate the switch logical SA6
| SA5,6,7,20..30 ! activate the logical switches
|           ! 5,6,7 and 20 to 30
| L3 = \L3 ! inverse the state of L3

```

What precedes is valid for the modification of all boolean variable of the type A, O, I, L, SA, I/O mnemonic, or other.

In a general way, if X and Y are both boolean variables, the writing of the following actions :

```

| X        ! set X to 1
| \X      ! reset X to 0
| X = Y    ! X takes the state of Y
| X = \Y   ! X takes the inverse state of Y
| X = \X   ! the state of X is reversed (\X
|           ! is placed in X)

```

In an affectation of boolean values with the sign =, the first member is a variable, and the second is an expression, that can have the syntax of the expressions described higher. For example, the following action is correct :

```

| X = L1 & L2 & (R5 > 1000)

```

Finally, the preset indexed variables have the particularity that they can be modified globally :

```

| L1,2,3   ! the L1 flags, L2 and L3 are set to 1
| \SA10..20 ! SA10 to SA20 are reset to 0
| A1,2,3,10..30 ! the A1 alarms, A2, A3, and A10
|           ! to A30 are set to 1
| \O0..O100 ! the O0 to O100 outputs are reset to 0

```

Actions on numeric variables

```

| R5 = 6    ! the value 6 is loaded
|           ! into the numeric register R5
| R6 = Clock - 1000 ! the value Clock - 1000
|           ! is loaded in the numeric register R6
| DT6 = R8 * 5    ! the result of the
|           ! multiplication of 5 by the content of the
|           ! R8 register is
|           ! loaded in the
|           ! DT6 temporization
| inc R3      ! increment of R3
|           ! equivalent in R3 = R3 + 1
| R7 = R7 + R8 ! addition of the content of the
|           ! R7 registers and R8, the result is charged
|           ! in R7
| inc R7, R8  ! idem previous line,
| dec R3     ! decrement of R3
|           ! equivalent to R3 = R3 - 1

```

Reset to 0 of variables

The boolean variables can be reset to 0 (false) by an instruction of the type (to see higher) :

```

| \L5      ! L5 = false

```

By extension, the same syntax can be used for the reset to 0 of the numeric variables or characters strings.

Examples

```

| \R5      ! R5 = 0
| \G3      ! G3 = 0
| \R5,10,20..30 ! R5, R10, R20 to R30 = 0
| \C1..C5,C10 ! C1 in C5, C10 empty (= "")
| \ANO4,5,6 ! ANO4 TO 6 = 0

```

Call of functions blocks or functions without return value

The call takes place while making precede the block of the keyword "DO."

Examples

```

| do CopyR( R100, R200, 20)      ! copy the registers R100 to R119 toward R200 to R219
|                               ! (call of function without return value)
|
| do BlockX( In1: = 3, In2: = 10 ) ! call of the BlockX function block with the In1 and In2 parameters
| R10 = BlocX.Out                ! copy of the Out output variable of BlockX in R10

```

6.8.4 Edge detection

The expressions described to the previous paragraph are true or false according to their state. It means that a condition as :

```
!SQ10          action1
```

is true as long SQ10 is to 1, false otherwise. Action1 is done as long as SQ10 is to 1, and this at every execution of the sequential.

If the action is a simple change of variable state, it doesn't pose a particular problem : the fact to force a variable in a state where it was already has no consequence.

On the other hand, if the action is a call of function, for example, it can be bothersome, especially if this call must be made once as the condition becomes true.

In some cases what one wants to test, is not the state of a condition, but its change of state : one speaks of "edge detection" :

- rising edge : change of a condition from 0 to 1

```

1          +-----
0 -----+

```

- falling edge : change of a condition from 1 to 0

```

1 -----+
0          +-----

```

- edge : change of state of a condition.

Three functions are available in Apigraf language for the detection of edge on a boolean expression :

- DFM (< boolean expression >)

detection of rising edge. The condition is true if the expression changes from 0 to 1.

- DFD (< boolean expression >)

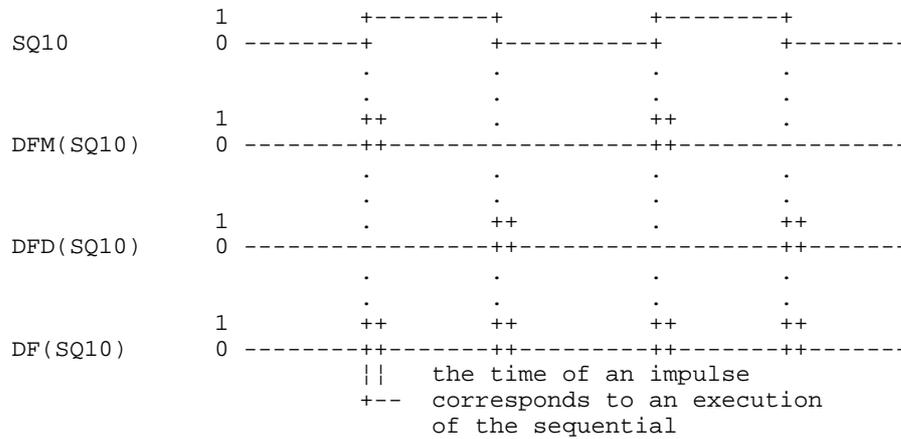
Detection of falling edge. The condition is true if the expression changes from 1 to 0.

- DF (< boolean expression >)

Detection of rising or falling edge (change of state). The condition is true if the expression changes state.

The expressions starting with one of these functions are only active during the duration of only one execution of the sequential.

Here is the chronogram of a condition for example without then with detection of edge:



Examples

```

!DFM(SQ10)                    ! condition true (1) if SQ10 changes
!                                ! from 0 to 1
!DFD(L10 & (R10 > 0)) ! condition true (1) if the expression
!                                ! L10 & (R10 > 0) changes from 1 to 0
!DF(SQ10)                    ! condition true (1) if SQ10 changes
!                                ! of state
  
```

Some precautions must be taken concerning the use of these functions of edge detection. In particular, they must be called to every scan of the sequential, in order to be able to compare the value of the present scan to the one of the previous scan. It implies the following rules (their failure to respect can entail erratic workings) :

- The functions of edge detection must not be used in GRAFCET steps, but only in combinative (outside of the sequences). It is necessary to notice that in a sequence, this kind of detection can always be done by adding a step of sequence for example, or by associating the detection of edge in combinative (outside of a GRAFCET sequence) to a flag tested in the step.
- To be valued correctly, these functions must not be used in the middle of a condition, but always to the beginning of the condition (because of the mechanism of optimization of the conditions evaluation, with which the evaluation is stopped as soon as the final result is certain). One only of these functions can appear therefore in a condition.

A detection of edge must always be placed at the beginning of line.

The following uses are either advised against or forbidden :

- In a SFC sequence, or an action, or a named block sporadically.
- Inside of a condition

```

!DFM(L1) # DFM(L2) # DFM(L3)
!                    L10
  
```

If a rising edge is detected on L1, DFM(L2) and DFM(L3) are not valued because the result of the condition is inevitably true. At the time of the following scans, the detection won't be made correctly.

The following writing would be correct :

```

!DFM( L1 # L2 # L3 )
!                    L10
  
```

...but doesn't give the wished result : for example if L1 is already to 1, the change of L2 from 0 to 1 doesn't provoke edge detection, since the expression L1 # L2 # L3 doesn't vary (it stays at 1).

In a case as this one, the solution would be to write :

```

!DFM(L1)    L10
!DFM(L2)    L10
!DFM(L3)    L10
  
```

6.8.5 Boolean conditions evaluation

The boolean expressions are valued from the left toward the right. As soon as the result of an expression is known in a certain way, the evaluation is stopped. This mechanism of optimization (common to the other textual languages of Optima PLC) permits to increase considerably the speed of execution of the sequential.

6.8.6 Numeric operations

A complete range of boolean, numeric operations or on characters strings are available, what permits to do complex enough operations directly in the PLC program.

ATTENTION

The sign = has two different roles following the place where it is written.
in a condition it means the comparison :

```
!R5 = R6          action1 ! if R5 is equal to R6
!                  ! then make action1
```

in an action it plays the role of affectation (writing) :

```
!condition1      R5 = R6      ! if condition1 is true
!                  ! then the content of R6 is
!                  ! affected (written in) in R5
!R5 = R6          R7 = R8      ! if the registers R5 and R6
!                  ! are equal, then copy
!                  ! R8 in R7.
```

Operations on numeric variables

+	addition of two values
-	subtraction of two values
*	multiplication of two values
/ or div	division of two numeric values
mod	modulo

Examples

```
! R20 = R6 + 4
! R1 = R6 * R7
! R2 = R8 R9 div
! R2 = 10 / 100
! R2 = real_to_dint (10.0 / 100.0)
! RR3 = 10 / R100      ! OK, but error if R100 = 0
!                      ! (real result)
```

Operators Inc and Dec

The following operators act on integer numeric variables:

inc	increment
dec	decrement

These operators act directly on the content of the variables and modify the value of it. Their use increases the speed of execution of the PLC program.

Examples

```
! inc R3          ! equivalent to R3 = R3 + 1
! dec R4          ! equivalent to R4 = R4 - 1
! inc R5, 1000    ! equivalent to R5 = R5 + 1000
! dec R6, R7+R8   ! equivalent to R6 = R6 - (R7+R8)
```

Comparison of numeric values

=	equality
<>	different
<	lower
>	superior
<=	lower or equal
>=	superior or equal

Examples

```
!(R1 < R2 + 3) action1 ! action1 activated if
!                                     ! R1 is bigger than (R2+3)
```

Scientific functions on numeric values

The standard functions available in the languages of the norm IEC 61131-3, those definite in the libraries Optima PLC and those definite by the user can be called in an expression. The types of data must be respected.

6.8.7 Writing on output text file

The "Write" and "WriteLn" instructions permit to send the text on the output text files present in the system (printers or files on disk).

The writings are always done in ASCII, at the end of the output (at the end of file if the output is a disk file). In addition to "Write", "WriteLn" does a jump to the following line.

Both instructions accept in brackets a continuation of parameters, of which the first is the name of interface (identifier) of the output (for example LPT1).

Example

```
!L10 WriteLn (LPT1, R1, Tab, R2, Tab, R3)
! \L10
```

When L10 changes to 1 (true), the following output is done on LPT1 (1st parallel printer) : value of the R1 register, one tabulation, value of the R2 register, one tabulation, finally the value of the R3 register. A return to the following line is done then (sequence CR LF).

The numeric values can be written while specifying a number of characters ; some spaces are added then on the left to get this number of characters if necessary.

Example

```
! WriteLn (LPT1, R1:10, R2:10)
```

Here, the output of R1 is done on 10 characters, then R2 on also 10 characters. Both values are stalled on the right, completed therefore on the left by spaces.

6.8.8 Grafcet instructions

A structure exists to define some sequences in the language.

The numbers of STEP of sequence are stored in variables of "SequenceApigraf" type, that must be defined in the global or local declarations (there are not predefined G variables like in Apigraf IP).

At one instant during the execution, only one step can be active for each sequence.

6.8.8.1 Declaration of a sequence variable

In the global variables (or local if it is about a function block), insert a declaration of the type :

```
G10 : SequenceApigraf;
```

The SequenceApigraf type is defined in the System library, that must be loaded therefore.

6.8.8.2 Definition of the code of a sequence

The syntax of beginning of a sequence (in the sequential) is the following:

```
! SEQUENCE xxx
```

where "xxx" indicates the name of the variable of the sequence. The word SEQUENCE must always be placed at the beginning of line.

6.8.8.3 Definition of a sequence step

the "STEP y" instruction starts the zone of the program where is going to be written the conditions / actions associated to this step of the GRAFCET current sequence :

```
| STEP y
```

"y" indicates the number of the step. It is an integer from 0 to 255.

After the STEP instruction are placed the boolean instructions of the conditions / actions type. The conditions are only tested when the sequence is in the corresponding step.

Just after the STEP instruction, a continuation of actions can be defined without condition : so that these actions are executed, it is sufficient that this step of sequence is active.

The unconditional actions placed directly after the STEP word are executed only one time, when the step becomes active. It permits to load a temporization, for example, solely at the time of the entry in the step.

When it is placed at the beginning of line, the STEP word indicates the beginning of a new step. If it is indented, it is about a jump instruction to another step of the sequence (to see example below).

6.8.8.4 End of a sequence

When all steps of sequence have been written, the end of the sequence is defined by the instruction "END" :

```
| END xxx
```

where "xxx" must be the same name that in the instruction SEQUENCE xxx of sequence beginning.

The END word must always be placed at the beginning of line.

6.8.8.5 Jump to a step of the sequence

When it is about passing from a step to another in a same sequence, the STEP instruction must be used, from inside a TRANSITION part.

Example

```

|....
|!-----
|STEP 4
|           KA1           ! motor starting (action)
|
|transition
|           ! beginning of the part transition
|           ! of the STEP 4 toward the other STEPS
|L5 & L10           STEP 5           ! jump from the step 4 to the step 5
|!-----
|STEP 5
|....

```

Explanation

In this sequence, (L5 & L10) constitute the transition (receptiveness) to go from the step 4 to the step 5.

The TRANSITION part can contain several transitions. However, if an empty transition is defined (no condition, only the destination STEP is indicated), only one transition is allowed then.

6.8.8.6 Examples of sequences

The general syntax of a sequence is the following:

```

|!-----
|G0 SEQUENCE           ! Beginning of the G0 sequence
|!-----
|STEP 0           ! beginning of the step 0
|...
|transition
|...

```

```

!-----
!STEP 1          ! beginning of the step 1
!...
!transition
!...
!-----
!STEP 2          ! beginning of the step 2
!transition
!...
!-----
!END G0          ! End of the G0 sequence
!-----
!-----
!G1 SEQUENCE    ! Beginning of the G1 sequence
!-----
!STEP 0
!...
!transition
!...
!-----
!STEP 1
!...
!transition
!...
!-----
!END G1! End G1 sequence
!-----

```

Example

```

! Blinking straight if stop
!Stop          \G1          ! straight sequence GRAFCET G1
!              \Lamp        ! extinction lamp.
!-----
!G1 SEQUENCE    ! beginning of the G1 sequence
!-----
!STEP 0
!transition
!Walk & \Stop
!              STEP 1      ! passage to the following step
!-----
!STEP 1
!              Lamp        ! ignition lamp
!              DT0 = R0     ! temporization lamp lit
!transition
!DT0 = 0        STEP 2     ! passage to the following step
!-----
!STEP 2
!              \Lamp        ! extinction of the lamp
!              DT0 = R1     ! temporization extinct lamp
!transition
!DT0 = 0        STEP 0     ! return in the beginning
!-----
!END G1          ! end of the G1 sequence
!-----

```

6.8.8.7 Transposition of a Grafcet in Apigraf language

```

! if emergency stop, stop the G1 sequence
EmergencyStop
  \G1

!-----
SEQUENCE G1
!
!-----+-----+
STEP 0
transition
SB1 STEP 1 ! if SB1 go to step 1 SB1 - + - !
!
!-----+-----+
STEP 1
YV3 ! opening of YV3
transition
SQ3 STEP 2 ! if SQ3, go to step 2 SQ3 - + - !
!
!-----+-----+
STEP 2
YV4 ! opening of YV4
transition
SQ4 STEP 0 ! if SQ4, go back to step 0 SQ4 - + - !
!
!-----+-----+
END G1
!-----+-----+

+-----+
+-----+
Condition Action Commentary Equivalent GRAFCET

```

6.8.9 Differences with the original Apigraf language

A certain number of differences between the original Apigraf language and the language implemented in Optima PLC must be taken in account at the time of the portage of an application of Apigraf PLC toward Optima PLC.

In general, the names of variables or functions are anglicized systematically.

<p>Declarations</p>	<p>The declarations of mnemonics and variables must be done with the syntax IEC 61131-3 in the part declaration of the POU's concerned. The keywords TITLE, "MESSAGE", E/S, "USES", "SEQUENTIAL", etc. of the Apigraf language are not more usable here.</p>
<p>Digital Inputs/outputs E and S</p>	<p>The prefix of access to the inputs and outputs are modified like follows : Exxx -> lxxx Sxxx -> Oxxx ES[xxx] cannot be used anymore, instead use I[xxx] or O [xxx] The new prefixes now correspond systematically to the prefixes used in English.</p>
<p>Analogical inputs/outputs EAN and SAN</p>	<p>The prefix of access to the inputs and analogical outputs are modified like follows : EANxxx -> ANlxxx SANxxx -> ANOxxx AN [xxx] cannot be used anymore, use instead ANI [xxx] or ANO [xxx] Attention: the values are now of the LREAL type, and not of DINT type. The new prefixes now correspond systematically to the prefixes used in English.</p>
<p>InitSequentiel, SequentielInit</p>	<p>This variable indicating the first scrutation of the PLC is not in use anymore in Optima PLC. To do an initialization, a variable created with an initial value defined to 1 can be</p>

	tested then modified of manner similar to InitSequentiel : var InitSequentiel : bool:= true; end_var Other manner, a task executed in StartPLC phase can be used.
DT temporizations	The type of the standard temporizations is not anymore DInt, but TIME. It is counseled in the new applications to use the syntax of the norm of the T#xxxx type. To facilitate the portage of program existing Apigraf, it is allowed however to affect a numeric value directly in milliseconds to the temporizations, or to use such a value in comparison.
Grafcets	See farther the differences specific to the grafcet.
Horloge, Clock, HorlogeInterne, InternalClock	These variables are not more available. Use instead the Timer function (in microseconds).
Time variables	To use the variables : Year, Month, Day, DayOfWeek, Hours, Minutes, Seconds, WeekNb, DayOfYear. The variable FacteurDates and FacteurSecondes are available under the DateFactor and SecondFactor names with compatible values with those of Apigraf. It is counseled however to use the types of variables defined in the norm IEC 61131-3.
Division	In the norm, the symbols / and div are equivalent. In the Apigraf language, for compatibility with the previous applications, these symbols are differentiated : <ul style="list-style-type: none"> • / give a real result • div give an integer result if the operands are integer, real otherwise.
Numeric functions	Some particular functions have been modified in agreement with the norm : ArcTan -> ATan Tan (tangent function previously inexistent)
Functions CopyR, CopieL, to SommeR	These functions are anglicized: CopyR, CopyL, SumR. Attention: SumR now returns a value of LInt type.
Ecris, EcrisLigne	These instructions are always available in French, but it is counseled to use the english equivalents Write and WriteLn.
InclureApigraf, IncludeApigraf	The inclusion of file is not possible in Optima PLC. To get a similar structure, with possible parameters, the included file must be transformed in independent function.

Differences specific to the syntax grafcet

No G preset table	The defined table Gx in Apigraf doesn't exist in Optima PLC. To define an Apigraf sequence in a POU Optima PLC, the variable of sequence must be defined by the programmer, either in the global variables, either in the variables of the POU. The type of variable is "SequenceApigraf." var G10 : SequenceApigraf; end_var This declaration then permits to define a G10 sequence. The name of the variables is not necessarily of the Gx shape.
--------------------------	--

<p>Key words anglicized</p>	<p>The key words are all anglicized : PAS → STEP APPEL → CALL FIN → END The other key words (SEQUENCE, TRANSITION) are unaltered.</p>
<p>No option of stability research</p>	<p>This option doesn't exist for the moment in Optima PLC.</p>
<p>Test of the activity of a sequence</p>	<p>The G10 condition inserted in an expression permitted to test the activity of G10, that is to say to know if a step different that 0 were active. This possibility not being allowed anymore, it is necessary to use the ApiSeqActive function : ApiSeqActive (G10) returns true if the sequence is not at the step 0.</p>

6.9 Advanced programming

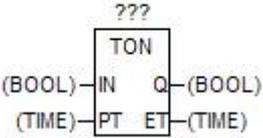
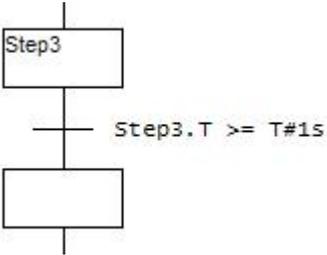
This chapter describes some particular techniques of current use.

6.9.1 Techniques of time management

With all system of automatic device development comes inevitably the question of the best manner to manage the time and the temporizations.

In Optima PLC, the management of the time can be done of different manners.

Methods of the norm IEC 61131-3

<p>Blocks standards TON, TOF, TP</p>	<p>These blocks are defined by the norm IEC 61131-3, and permit to manage some temporizations on rising or falling edges. These are function blocks, they must be declared therefore in the variables to be used in a block. Their use is possible in all languages of Optima PLC. TON is defined for example like follows :</p> 
<p>Elapsed time for a step ***.T</p>	<p>The elapsed time in a step of SFC diagram can be used in the following transitions, as in this example :</p>  <p>This syntax is defined in the norm IEC 61131-3. The time may come from an expression or variable of TIME type.</p>

Methods specific to Optima PLC

<p>DT temporizations</p>	<p>These temporizations are indexed variables of Time type, particular in this sense that since the loading with given one time, the value decreases with the elapsed time until reaching 0. It is sufficient to load a variable DT with one time therefore, then to wait that the value reach 0. Example in ST language (blinking of the L5 flag to 1 Hz) :</p> <pre> if DT5 = T#0 // is the time elapsed ? then L5 := not L5; // inversion of the value of the flag </pre>
---------------------------------	--

	<pre>DT5: = T#500ms // reloading of the temporization end_if;</pre> <p>These variables are inherited of the original Apigraf language. They can be used however in all available languages in Optima PLC.</p>																						
<p>Functions returning the current value of the time : Timer, Current_DT, Current_Date_Current_TOD</p>	<p>The management of the time can sometimes be achieved while memorizing the value current of the time, then while doing the difference between the current value and the value memorized. The functions Timer (Time) and Current_DT can be used for it.</p> <p>Attention however to the Current_DT function : it presents the advantage to return an absolute time, it is sensitive at a change of date and hour provoked by the user on the PC, what can distort the measures of time.</p> <p>The Timer function returns a time whose origin is indefinite (what permits to use it only in the differences of elapsed time), but that is not affected by a change of date and hour.</p>																						
<p>Time management variables</p>	<p>The following variables permit an advanced management of the time :</p> <table border="0"> <tr><td>Year</td><td>Current year</td></tr> <tr><td>Month</td><td>Current month</td></tr> <tr><td>Day</td><td>Day of the month</td></tr> <tr><td>Hours</td><td>Current hours</td></tr> <tr><td>Minutes</td><td>Current minutes</td></tr> <tr><td>Seconds</td><td>Current seconds</td></tr> <tr><td>DayOfYear</td><td>Number of the day in the year</td></tr> <tr><td>DayOfWeek</td><td>Number of the day in the week (0 = Sunday)</td></tr> <tr><td>WeekNb</td><td>Number of the week in the year</td></tr> <tr><td>DateFactor</td><td>Absolute numbering of the day (Apigraf compatibility)</td></tr> <tr><td>SecondFactor</td><td>Absolute numbering of the second (Apigraf compatibility)</td></tr> </table>	Year	Current year	Month	Current month	Day	Day of the month	Hours	Current hours	Minutes	Current minutes	Seconds	Current seconds	DayOfYear	Number of the day in the year	DayOfWeek	Number of the day in the week (0 = Sunday)	WeekNb	Number of the week in the year	DateFactor	Absolute numbering of the day (Apigraf compatibility)	SecondFactor	Absolute numbering of the second (Apigraf compatibility)
Year	Current year																						
Month	Current month																						
Day	Day of the month																						
Hours	Current hours																						
Minutes	Current minutes																						
Seconds	Current seconds																						
DayOfYear	Number of the day in the year																						
DayOfWeek	Number of the day in the week (0 = Sunday)																						
WeekNb	Number of the week in the year																						
DateFactor	Absolute numbering of the day (Apigraf compatibility)																						
SecondFactor	Absolute numbering of the second (Apigraf compatibility)																						

Methods specific to Optima PLC, based on UTC (Coordinated Universal Time)

<p>Functions returning the current value of the time Current_DT_UTC, Current_Date_UTC, Current_TOD_UTC</p>	<p>That functions can be used as those above, but give a time independant from the location and the position in the year (daylight), and are ideal for absolute time-stamping purposes.</p>																		
<p>UTC variables</p>	<p>The UTC structure give the UTC current time :</p> <table border="0"> <tr><td>UTC.Year</td><td>Current year</td></tr> <tr><td>UTC.Month</td><td>Current month</td></tr> <tr><td>UTC.Day</td><td>Day of the month</td></tr> <tr><td>UTC.Hours</td><td>Current hours</td></tr> <tr><td>UTC.Minutes</td><td>Current minutes</td></tr> <tr><td>UTC.Seconds</td><td>Current seconds</td></tr> <tr><td>UTC.DayOfWeek</td><td>Number of the day in the week (0 = Sunday)</td></tr> <tr><td>UTC.DateFactor</td><td>Absolute numbering of the day (Apigraf compatibility)</td></tr> <tr><td>UTC.SecondFactor</td><td>Absolute numbering of the second (Apigraf compatibility)</td></tr> </table>	UTC.Year	Current year	UTC.Month	Current month	UTC.Day	Day of the month	UTC.Hours	Current hours	UTC.Minutes	Current minutes	UTC.Seconds	Current seconds	UTC.DayOfWeek	Number of the day in the week (0 = Sunday)	UTC.DateFactor	Absolute numbering of the day (Apigraf compatibility)	UTC.SecondFactor	Absolute numbering of the second (Apigraf compatibility)
UTC.Year	Current year																		
UTC.Month	Current month																		
UTC.Day	Day of the month																		
UTC.Hours	Current hours																		
UTC.Minutes	Current minutes																		
UTC.Seconds	Current seconds																		
UTC.DayOfWeek	Number of the day in the week (0 = Sunday)																		
UTC.DateFactor	Absolute numbering of the day (Apigraf compatibility)																		
UTC.SecondFactor	Absolute numbering of the second (Apigraf compatibility)																		

6.9.2 Display of information, of errors, of confirmation demands

The library Optima PLC "Messages" includes the functions and function blocks permitting to display an information text or a demand of confirmation.

Some parameters are common to these functions :

Name	Type	Description
LogEvent	BOOL	This parameter can be put to 1 so that the system records the message in the log. At 0, the message is not recorded. The value is by default 1.
HelpTopic	STRING	This parameter permits to define the access to a help concerning the displayed message.

		<p>The value must be not empty so no help is available. If a help is available, the value must correspond to a topic of a .chm file situated in the directory of the application (if possible UserMessages.chm). Possibly, the topics of help can be distribute in several .chm files.</p> <p>When the message is displayed, the button Helps can be clicked to provoke the opening of the help window.</p>
--	--	---

6.9.3 Use of the samples tables standards

The "samples tables" are interactively definite structures in the options menu of the workshop Optima PLC. Their interest resides mainly in the fact that they are to the basis of the displays of curves in the supervisory Optima PLC.

Once the table "TabData" (for example) defined in the list of the measure tables, it is possible to use the following expressions in a program :

Expression	Description	Type
TabData	Reference to the table, used in some functions	UDINT
TabData [index]	direct access to a value of the table	DINT
TabData.Size	Capacity of storage in the table	DINT
TabData.Count	Number of values currently stored in the table	DINT
TabData.Min	Minimal value placed in the table	DINT
TabData.Max	Maximal value placed in the table	DINT

The standard library "TabSamples" contains a certain number of functions for manipulation of the measure tables.

6.9.4 Use of the output text files

The "output text files" are interactively definite structures in the options menu of the workshop Optima PLC.

To the difference of the general files manipulated with the library "FILES", these files permit to achieve some outputs in text files solely. However these outputs are facilitated distinctly, in particular thanks to the functions of the TEXTOUT library.

Once the output file defined, this one can be used in the program, in particular with the expandable functions (the number of parameters is variable) TOWrite and TOWriteLn. The variants TOWriteCSV, TOWriteLnCSV permit to write some lines directly to the CSV ("comma separated values", values separated by commas) format. The variants TOWriteTab and TOWriteLnTab permit to write directly to the TSV ("tab separated values", values separated by tabulations) format. The formats CSV and TSV are generally used to generate spreadsheet files in an open format.

The first parameter to pass for each of these functions is the file itself (use the name specified in "interface file"). Then, as many parameters as wished can be pass, each with any simple type. The TOWriteLn functions add a line jump following the printed fields.

The text sent on these files of output can be consulted during the execution of the runtime, with the option [Text outputs](#)^[33].

7 "On line" modifications

Optima PLC is able to modify an application under working without interrupting it. Some rules, described below, must be however respected strictly.

7.1 Application

7.1.1 Modifications of the program of the application

The POU's of an application can be modified on line. An option of the menu Compilation permits this type of modification ([Compilation and on line modification](#)^[18]).

However some important restrictions can be imposed according to the type of POU. A general rule is that the authorized modifications must not influence on the global structure of the application data. Thus, it is not possible to add or to suppress some variables, or to modify the size or the format of it.

This rule is relatively restraining, because it implies that it is impossible to add or to withdraw blocks of functions. In the same way, the "static" variables of the blocks of functions (input, output and internal variables) cannot be modified. But the modification of the code executed in the POU's is well often extensively sufficient at the debug time.

Here is for every type of POU the applicable restrictions.

	Type of POU	Authorized modifications	Forbidden modifications
	Task	Modification of the trigger and the period	Modification of the list of the programs called in the task. Modification of the priority of the task.
	Global variables	Addition and modification of declared variables with AT: it is in general about the definition of mnemonics. New mnemonics can be added without problem. Modification of the name of the normal global variables. Passage of normal variables to "retain" and vice versa. The modification is not however taken in account before the recompilation and normal execution of the application.	Modification of the normally declared global variables (type, addition or deletion of variables, etc...)
	definitions of types	Modification of the elements of an enumerated type.	No other modification is allowed, unless it is about a type non used in the application.
	program block	See function block	See function block
	function block	Modification of the temporary variables (VAR_TEMP). All modifications of the code, except in the case of the SFC (graphic and textual). Modification of the code and the call mode of the actions in SFC. Inhibition and de-inhibition of call of actions in SFC. Addition of ONE action in the empty SFC steps.	Modification of the input variables, output and internal variables ("VAR"). Modification of the structure of a SFC diagram (addition or deletion of steps, transitions and calls of actions).
	function	All modifications	
	subgroup	All modification concerning the order or the hierarchy of the POU's	Deletion of function blocks or program used in the application.

7.1.2 Modes of loading of the modifications

The modifications done and compiled during the working of the application are placed in a code file of .prg extension, as the standard module of application OptimaPLC.prg. This type of modification module is named OPLCXXXX.prg, XXXX being already replaced by the first numeric value available among the preset files, starting with 0000.

The runtime takes in charge the demand of loading of the modification module done by the workshop Optima PLC. A previous confirmation (that can be disabled) is necessary.

The loading, once done, is signalled in the events.log file. The code charged in memory is executed then from the two OptimaPLC.prg modules and OPLC0000.prgs (for example).

Loading of a new on line modification

If a precedent modification module was already present at the time of the loading of a new module, the first one is withdrawn from the memory after the toppling over toward the new module, in this manner one module of modification at the maximum is charged at one time. Indeed, a module takes the set of the modifications done since the starting of the application.

Starting outside of the workshop of an application modified

If the modified application is started then without passing by the workshop Optima PLC, the OptimaPLC.prg module is first of all loaded normally, then the last module of modification is loaded too. If this module has been suppressed by hand in the meantime on the disk, a confirmation is asked to start without the on line modifications, or to stop the starting.

Automatic deletion of the on line modification modules

At the time of a normal compilation producing an OptimaPLC.prg file, the OPLCXXXX.prg files possibly present in the directory are suppressed.

7.2 Modifications of the supervisory

The supervisory forms can be modified and can be tested without any problem during the working of the application.

However, a form is taken into account by the application at the next loading of this one.

8 Examples

The goal of these applications is to permit a hold in hand of the software Optima PLC by simple and playful examples.

8.1 Creation of a program

To create an application on Optima PLC, it is first of all necessary to create a projectl (see Chapter: Creation of a project). Call this program for example "Motor Starting."

Creation of the POU's

The POU's (Programming Organization Units) are the objects that permit to create :

- Task: POU of cadencement of the program blocks execution
- Global variables: POU of variables declarations for the set of the POU's
- Types: POU of type variables declarations for the set of the POU's
- Block of program: Main programming POU
- Block of function: subroutine, with or without return value, that memorizes the state of its internal variables between 2 calls, so for a same set of input parameters the result returned by this block can be different (example of the PID: for a same order the exit of this regulation block evolves until it reaches the order)
- Function: subroutine of which the returned value remains identical for a same set of input parameters (no memorization of the variables interns)
- Subgroup: POU of library structuring

You will notice that some POU's already exist in the library, they compose the minimum program.

To begin, it is necessary for you to declare the variables of your system. For that, create a POU of global variables that you will name "Variable".

8.2 Example 1: Motor starting

8.2.1 Working

This example is based on the starting of a motor.

The support on "marche" (walk) provokes the rotation of the motor and allows the auto-maintenance of the system to speed up. A support on "arrêt" (stop) permits to stop the system.

In boolean the equation of the motor is written :
Fonctionnement:= (Marche+Fonctionnement). not Stop

(fonctionnement = working)

This equation is programmed therefore with the different programming language existing in Optima PLC

1. ProgrammeST: Programming in structured language

The ST language (Structured Text) is a high-level textual language dedicated to the applications of automation. This language is used mainly to describe the complex procedures, programmable with difficulty with the graphical languages.

2. ProgrammeFBD: Programming in function block diagram (FBD)

The FBD language (Function Block Diagram) is a graphical language. It allows the construction of complex equations from the standard operators, functions or functional blocks.

4. ProgrammeLD: Programming in Ladder language (or contact language)

The LD language (Ladder Diagram) is a graphical representation of boolean equation combining contacts for the inputs and of relays for the outputs.

5. ProgrammeSFC: Programming in Language SFC

The SFC language, (Sequential Function Chart) or GRAFCET, is a graphical language used to describe the sequential operations. The process is represented like a continuation of steps bound between them by transitions.

8.2.2 Declarations

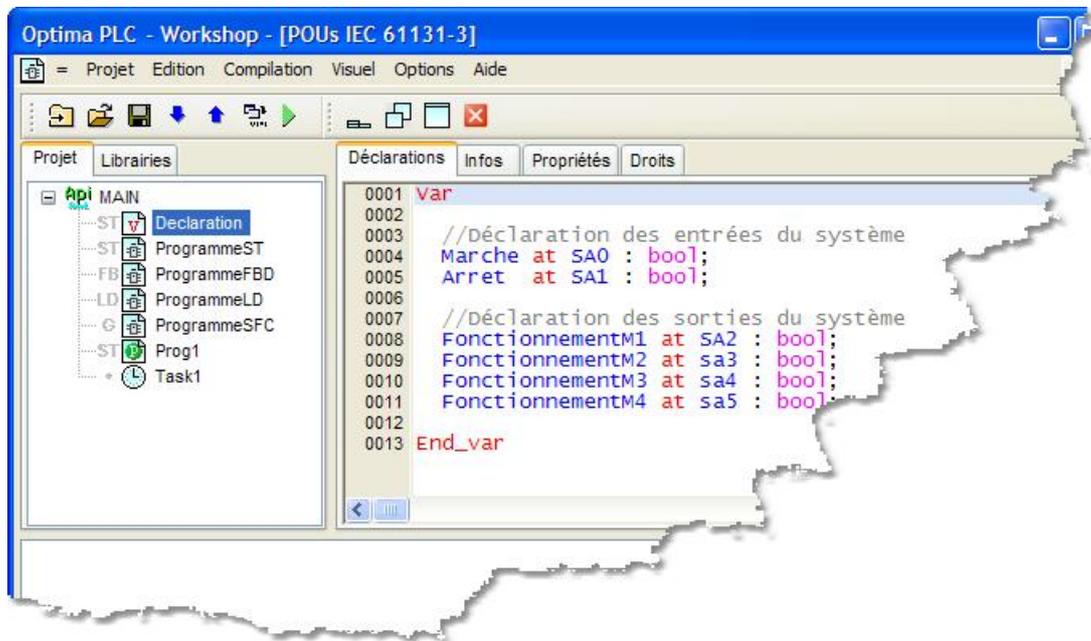
It is necessary in a first time to declare the variables that will be the inputs and the outputs of the programs.

For that create a POU, choose the option to "Create global variables" while making a right click on MAIN in your tree.

It is now sufficient to declare all necessary variables.

For example :

- Marche (Walk) corresponds to the denomination of the variable
- SA0 corresponds to its internal preset variable eventually used (non obligatory)
- bool corresponds to its type, here it is a boolean.



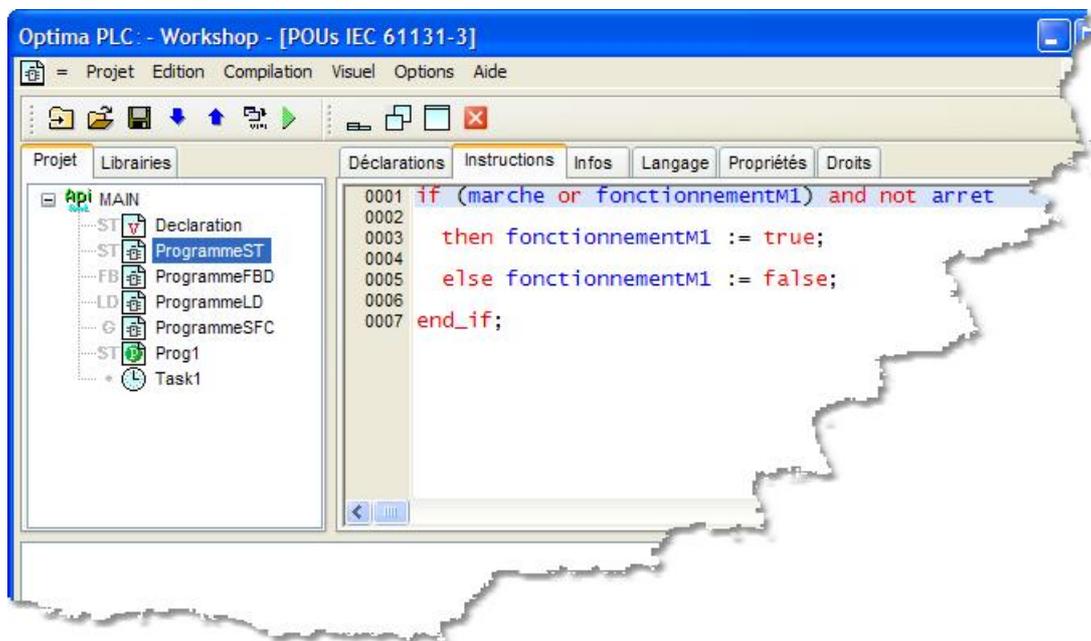
8.2.3 ProgrammeST

The first program is a program in structured language.

Make a right click on MAIN, choose to "Create a block of function", call it ProgrammeST, choose the option ST language, then validate.

It is necessary now to program the boolean equation of [FonctionnementM1](#)³⁸⁰ (working).

Hit the following instructions as indicated on the copy of screen.



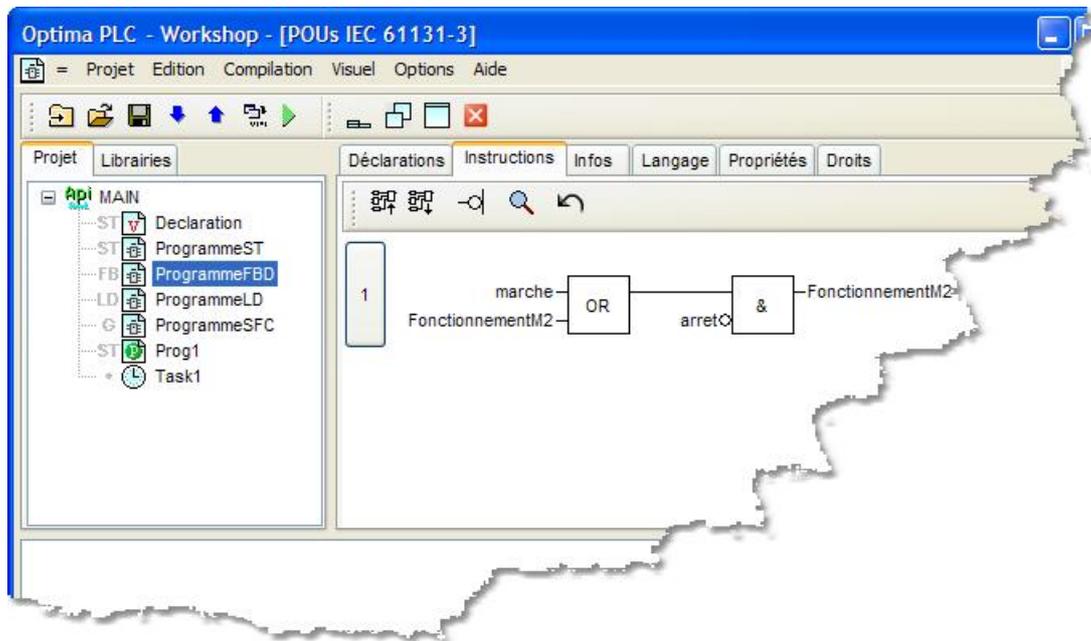
8.2.4 ProgrammeFBD

The following program is a program in FBD language.

Make a right click on MAIN, choose to "Create a block of function", call it ProgrammeFBD, choose the option FBD language, validate then.

It is necessary now to program the boolean equation of [FonctionnementM2](#)³⁸¹ (working).

Hit the following instructions as indicated on the copy of screen.



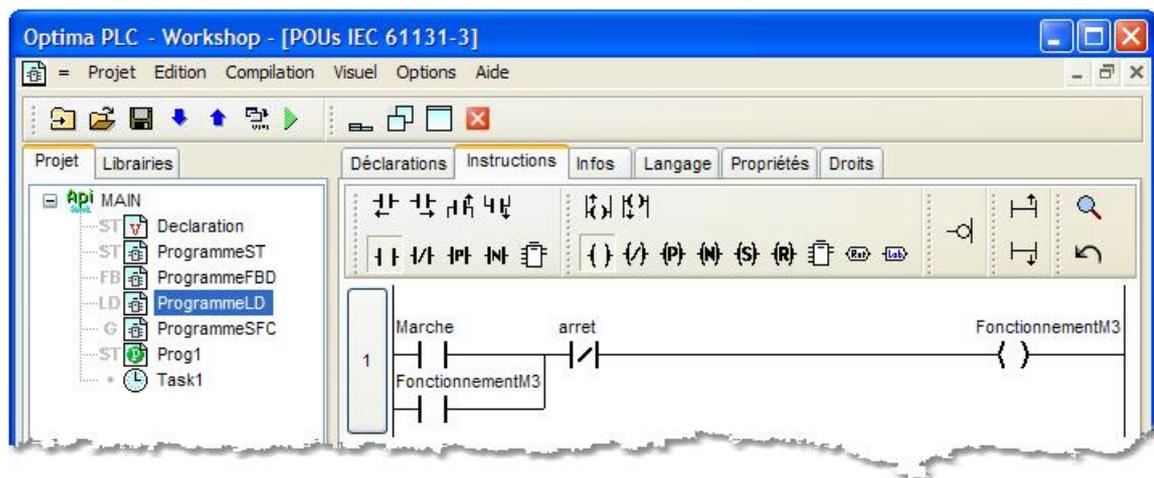
8.2.5 ProgrammeLD

The following program is a program in Ladder language (contact language).

Make a right click on MAIN, choose to "Create a block of function", call it ProgrammeLD, choose the option LD language, validate then.

It is necessary to program the boolean equation of [FonctionnementM3](#)^[380] (working).

Hit the following instructions as indicated on the copy of screen.



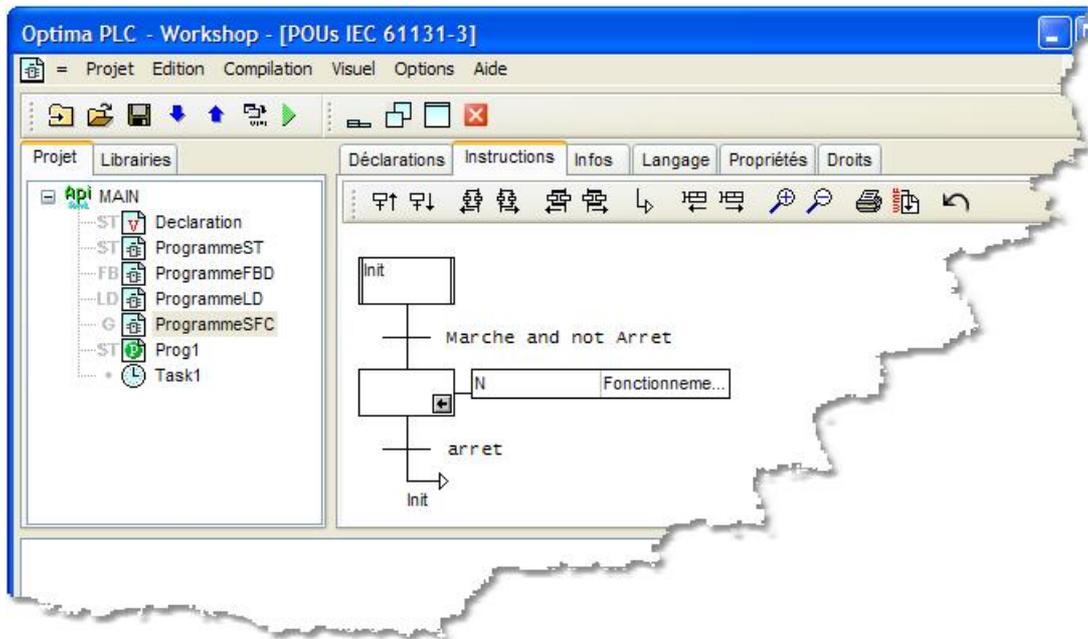
8.2.6 ProgrammeSFC

The following program is a program in SFC (GRAFCET) diagram.

Make a right click on MAIN, choose to "Create a block of function", call it ProgrammeSFC, choose the option SFC language, validate then.

It is necessary to program the boolean equation of [FonctionnementM4](#)^[381] (working).

Hit the following instructions as indicated on the copy of screen.



8.3 Example 2: Automatic door

8.3.1 Working

Description of the mode of working :

The user presents himself in front of the door, he has possibility to open the door with two different manners :

- he possesses the key
- he enters the entry code on the keyboard

The opening of the door is made until it reaches the end sensor of opening. It closes again itself at the end of a temporization T1. The person has a button inside to open the door at any time.

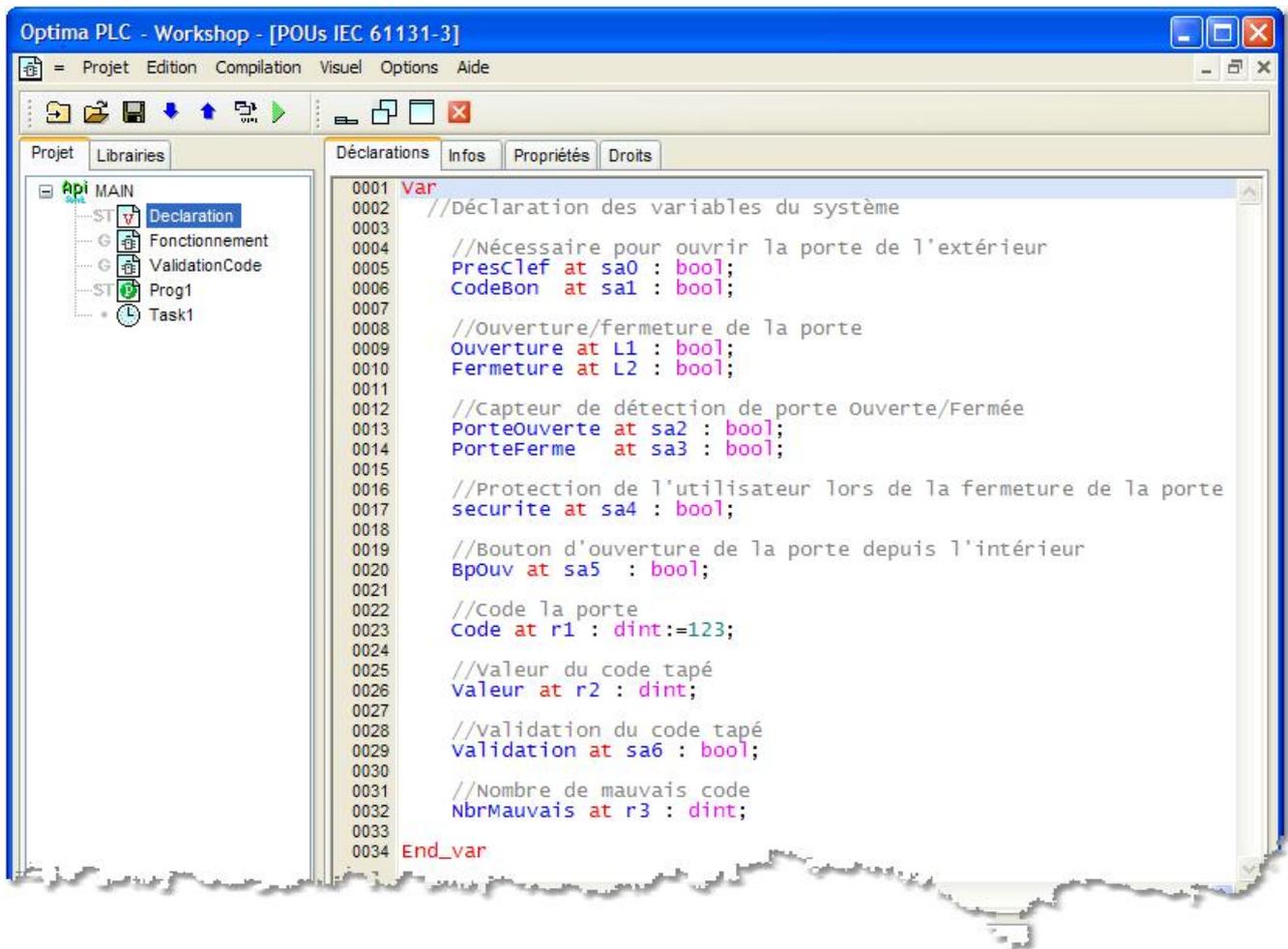
When the person leaves the surrounding wall, the door closes again itself at the end of a temporization T2.

By measure of security, a pressure sensor is installed on the door. This sensor permits to open the door if something blocks the door when closing.

The user has three trials to enter the code that opens the door. After three trials, a new trial will only be possible after a temporization T3.

8.3.2 Declarations

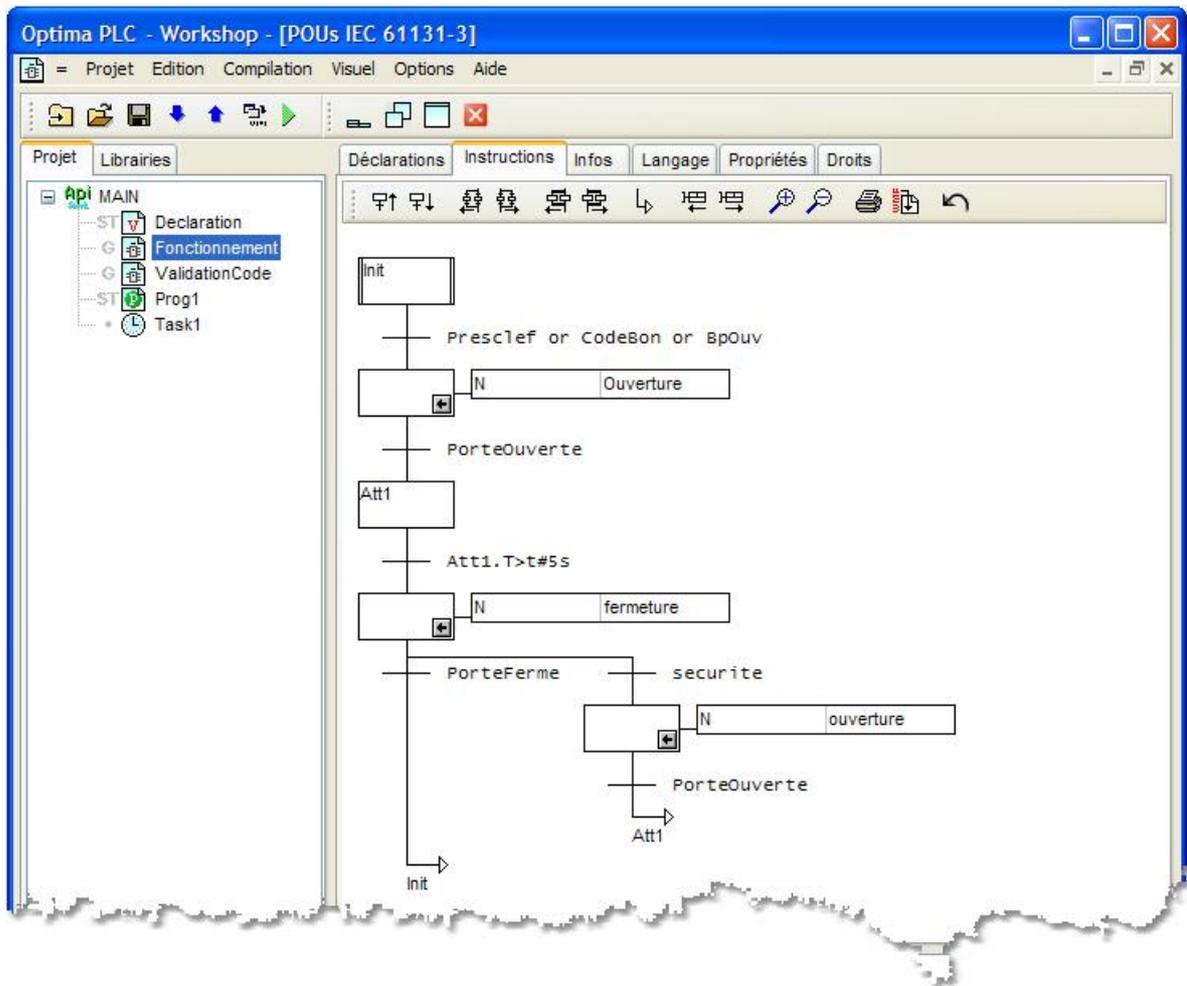
Declaration of the variables of the system.



8.3.3 Working

Here is the program of the door, it is a basic program of opening and closing of automatic door.

Except that at the time of closing, if someone or something blocks the closing of the door, a security sensor permits the reopening of the door.



8.3.4 Validation of the code

It is now necessary for us to create a POU which permits a management of the door security code.

- Create a function block that you call ValidationCode and choose the SFC language.
- You first of all have an initial step, there we are going to initialize the variables of this program (for example the code of the door, here "123")

Make a right click on the step and choose to "Add an action", a window opens up "Parameters of action." Check the box "Edition of the action in situ in ST language" to permit the writing of the variables initialization. Validate.

- Click on the action box to edit the action in structured language :
- Hit the following sequence :

```
Code := 123; // initialization of the code
CodeBon:= False; // CodeBon to 0
```

- From this initial step, the program waits for the validation of the code hit, from then it is going to compare the value entered "Value" with the real value of the code "Codes."

It is therefore necessary to create one "or" following the case where the code is good or bad.

- Select the icon "alternative branch after". The "or" is created now, we are going to consider the branch of left as the validation of a GOOD code and vice versa for the one of right.
- Make a right click on the transition of left "Edit the transition in situ (ST.)" and to edit it :

```
Validation and (Value = UsersCode)
```

- Make a right click on the transition of right to "Publish the transition in situ (ST.)" and to publish it :

Validation and (Valeur < > UsersCode)

"=" in ST. the equality means
"<>" in ST. the difference means

- In a first time, we are going to be interested in the branch "codes good"

Select the following step, Make a right click here on this one and choose to "Add an action", check the box "Edition of the action in situ in ST language", Validate. This step permits to transmit information "good code" to the program. It also resets the number of bad codes.

The following transition will be validated if CodeBon = True, This transition is necessary to insure that information has been transmitted well to the program of the door.

- Finally, we are going to be interested in the branch "codes bad"

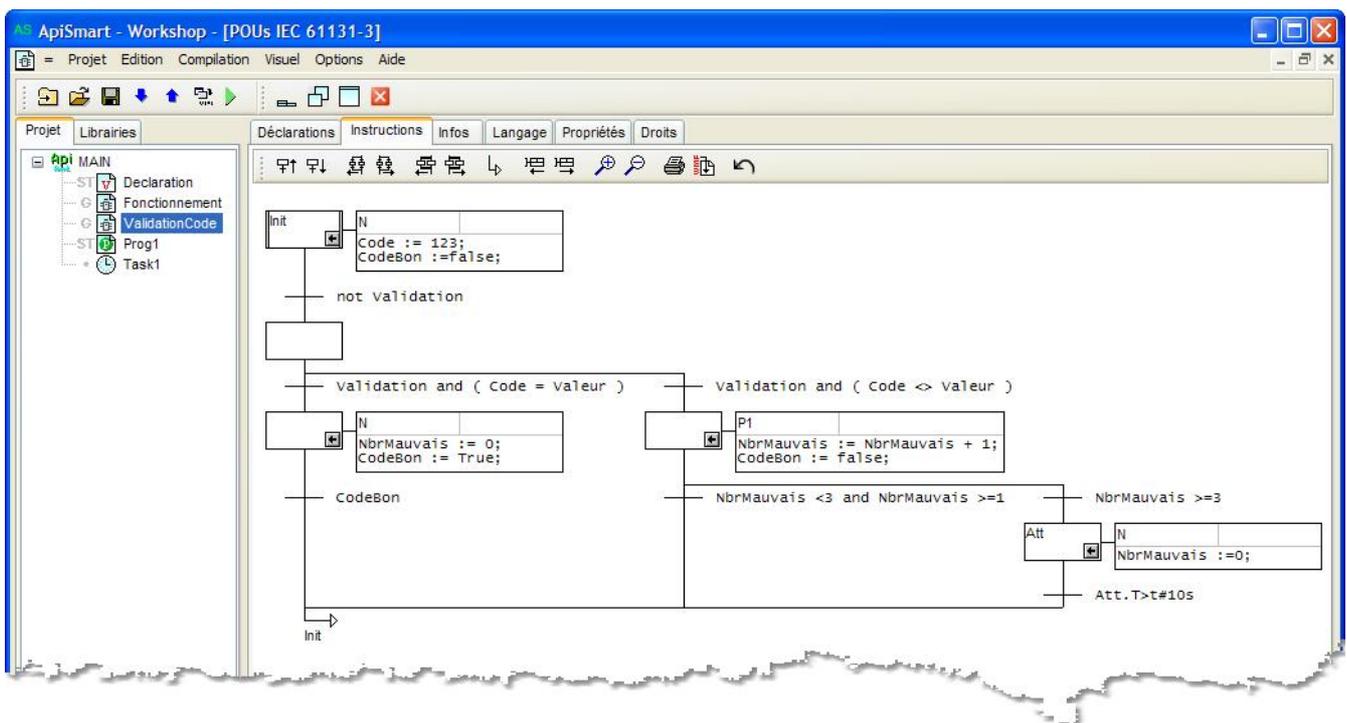
If the validation has been done and the entered code is not valid, it is necessary to increment the number of bad code. For that create a step that will permit the increment and that will return information CodeBon := False.

It is necessary to choose the P1 qualifier so that the action executes itself only one time from the moment the step is active.

There is now two possible cases once the step active : either the number of bad code is superior or equal to three, or it is lower to three.

It is necessary to create therefore one "or".

In the case where the number of bad codes is superior or equal to three, it is necessary to make the user wait.



8.4 Other Optima PLC examples

Several other examples are provided and are installed at the same time as optima PLC. They are accessible in the "Examples" directory situated under the installation directory. To see these examples, it is as possible to click on the button "Examples" in the window of project selection.

Curves

Demonstration of values recording and display of curves.

Elevator	Simple example of elevator simulation.
Crossing lights	Simple example of cycle of signaling of a crossroads.
Mixer	Cycle of management of a mixer.
ModbusExchanges	Demonstration of the Modbus library.
RegulPID	Demonstration of the PID regulation tools (compatible with the corresponding tools in Apigraf)
Carpet	Simple example of management of a graph of pusher
Tests	This index contains various test projects making each work one or several particular functions of optima PLC.

9 Drivers Optima PLC

The drivers constitutes the tie between Optima PLC and the input/output hardware.

This chapter describes the drivers available in standard.

9.1 Advantech

9.1.1 Card Advantech PCL-728

Version 1.0



This interface allows access to analog channels of the card PCL-728.

2 analog output channels are available (1 and 2).

The default configuration of the card gives an output voltage between 0 and +5 V.

9.1.1.1 Parameters

To configure the driver, select it in the list of the drivers then click on the button "Parameters."

There are three parameters but only the parameter "IO base address" is obligatory, the other have some default values :

- IO base address :

The IO address is defined on the card by the DIP switches (the proposed value corresponds by default to the configuration manufactures).

- Mode of every channel (configured on the card with JP1 and JP5 for the channel 1, JP2 and JP9 for the channel 2) :

1 for unipolar or 4 - 20 my,
2 for bipolar.

9.1.1.2 Usage

In the development workshop

The input/outputs must be declared either in the part "global Variables." The 2 analog outputs are of LREAL type.

Example

```
PCL728Channel1 ANO0 at: LREAL;
PCL728Channel2 ANO1 at: LREAL;
```

To use them in the program, it is possible to use the variable (ex: ANO0) or its mnemonic (ex: PCL728Channel1).

In unipolar, the range of the channels go from 0 to 4095, in bipolar of -2048 to 2047.

Example (channel 1 in unipolar and channel 2 in bipolar) :

```
PCL728Channel1: = 2000;
PCL728Channel2: = -400;
```

In the runtime Optima PLC

The inputs/outputs are displayable in the option "PLC - > Variable - > Analog output ANO."

9.1.2 Card Advantech PCI-1710

Version 1.0



This driver permits the usage of the multifunction boards Advantech PCI-1710.

The card has 16 analog inputs 12 bits with programmable gain, 2 analog outputs 12 bits, 16 inputs and 16 digital outputs. The apportionment is made of the following manner :

	First channel	Last channel
Digital inputs	0	15
Digital outputs	16	31
Analog inputs	0	15
Analog outputs	16	17

The driver provided by the builder for Windows can be installed beforehand in the goal to examine the good performance of the card with the provided utilities.

However, the installation of the Advantech driver is not obligatory to use the card with Optima PLC. The presence of this one as well as its base address are automatically determined by Optima PLC, that performs some direct accesses then to the hardware in order to permit the obtaining of real-time performances.

The card is also provided under three other versions :

Model	Difference in relation to the PCI-1710
PCI-1710L	No analog output.
PCI-1710HG	More elevated values (HG = High Gain) of gain.
PCI-1710HGL	More elevated values of gain and not of analog output.

9.1.2.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) on the card permit to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

- Frequency by channel

Number of acquisitions wished per second and by channel. Attention, this frequency, in particular if it is raised, cannot always be gotten in an exact manner. The log file mentions to the start of the application an approximation of the frequency if this one was necessary.

The reading of the FIFO of the card consumes a relatively elevated CPU time, a global frequency of acquisition too high risks therefore to penalize the foreground functions of the application (user interface), or even to block them. A global frequency of 50000 acquisitions per second, seem a reasonable limit. It is possible to modulate the frequency of the acquisitions with the functions provided by the PCI1710 library, that permits not to penalize the rest of the system permanently if an elevated frequency is only useful to precise moments.

- Number of values in sample tables

Size of sample tables attached to every channels, permitting to perform background acquisitions (see PCI1710 library).

- Scale of measure channels x to y

Scale and type of measure for the group of 2 channels specified.

- Reference voltage analog outputs

For every analog output, the reference voltage can be defined: 5V internal, 10V internal, or external tension. Attention: in the first two cases, the value placed in the analog output is understood respectively between 0 and 5.0 and between 0 and 10.0. In the case of the external reference voltage, the value must be between 0 and 4095.

9.1.2.2 Usage

In the development workshop

The inputs/outputs must be declared either in a part "global Variables." The analog inputs are of LREAL type.

Example

```
Channel1 ANI0 at: LREAL; // first analog input
Channel2 ANI1 at: LREAL; // second analog input

ChanOut1 ANO16 at: LREAL; // first analog output
```

To use them in the program, it is possible to use the variable (ex: ANI0) or its mnemonic (ex: Channel1).

The values are returned in Volts, while taking into account the scale.

In the runtime Optima PLC

The inputs/outputs are displayable in the menu "PLC - > Variable - > Analog input ANI" and "Analog output ANO."

Usage of the sample tables

The functions of acquisitions in tables are available while using the PCI1710 library. Data are stocked in these tables to the asked acquisition frequency.

The tables are named of the following manner :

PCI1710_ < Card ID > _ < Nb of channel >

Thus, the channel 10 of the card 0 will be stocked in the PCI1710_0_10 table.

The stocked values are integer, between 0 and 4095 if the channel is unipolar, between -2048 and 2047 if the channel is bipolar. For every table, the display coefficient is defined according to the acquisition range of the channel, what permits a display in V.

Besides, the "time" is stocked in a table :

PCI1710_ < Card ID > _Time

The stocked values are the indexes of acquisitions (0, 1, 2, etc.) and the coefficient is defined in order to permit a display in seconds.

9.1.2.3 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	No card detected PCI-1710 with the set card number
3	Incorrect frequency by channel (must be between 1 and 50000 Hz)
4	Incorrect values range for a channel group

9.1.3 Card Advantech PCI-1711

Version 1.0



This driver permits the usage of the multifunction boards Advantech PCI-1711.

The card has 16 analog inputs 12 bits with programmable gain, 2 analog outputs 12 bits, 16 inputs and 16 digital outputs. The apportionment is made of the following manner :

	First way	Last way
Digital inputs	0	15
Digital outputs	16	31
Analog inputs	0	15
Analog outputs	16	17

The driver provided by the builder for Windows can be installed beforehand in the goal to examine the good performance of the card with the provided utilities.

However, the installation of the Advantech driver is not obligatory to use the card with Optima PLC. The presence of this one as well as its base address are automatically determined by Optima PLC, that performs some direct

accesses then to the hardware in order to permit the obtaining of real-time performances.

The card is also provided under another version, PCI-1711L, not having an analog output.

9.1.3.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) of the card permits to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

- Frequency by channel

Number of acquisitions wished per second and by channel. Attention, this frequency, in particular if it is raised, cannot be gotten in an exact manner. The log file mentions to the start of the application an approximation of the frequency if this one was necessary.

The reading of the FIFO of the card consumes a relatively elevated CPU time, a global frequency of acquisition too high risks therefore to penalize the foreground functions of the application (user interface), or even to block them. A global frequency of 50000 acquisitions per second seems a reasonable limit. It is possible to modulate the frequency of the acquisitions with the functions provided by the PCI1711 library, what permits not to penalize the rest of the system permanently if an elevated frequency is only useful to precise moments.

- Number of values in sample tables

Size of the sample tables attached to every channel, permitting to perform background acquisitions (see PCI1711 library).

- Scale of measure channels x to y

Scale and type of measure for the group of 2 channels specified.

- Reference voltage analog outputs

For every analog output, the reference voltage can be defined: 5V internal, 10V internal, or external tension. Attention: in the first two cases, the value placed in the analog output is understood respectively between 0 and 5.0 and between 0 and 10.0. In the case of the external reference voltage, the value must be between 0 and 4095.

9.1.3.2 Usage

In the development workshop

The inputs/outputs must be declared either in a part "global Variables." The analog inputs are of LREAL type.

Example

```
Channel1 ANI0 at: LREAL; // first analog input
Channel2 ANI1 at: LREAL; // second analog input

ChanOut1 ANO16 at: LREAL; // first analog output
```

To use them in the program, it is possible to use the variable (ex: ANI0) or its mnemonic (ex: Channel1).

The values are returned in Volts, while taking into account the scale.

In the runtime Optima PLC

The inputs/outputs are displayable in the menu "PLC - > Variable - > Analog input ANI" and "Analog output ANO."

Usage of the sample tables

The functions of acquisitions in tables are available by using the PCI1711 library. Data are stocked in these tables to the asked acquisition frequency.

The tables are named of the following manner :

```
PCI1711_ < Card ID > _ < No of channel >
```

Thus, the channel 10 of the card 0 will be stocked in the PCI1711_0_10 table.

The stocked values are full, included between 0 and 4095 if the channel is unipolar, between -2048 and 2047 if the channel is bipolar. For every table, the coefficient of display is defined according to the range of acquisition of the channel, what permits a display in V.

Besides, the "time" is stocked in a table :

PCI1711_ < Card ID > _Time

The stocked values are the indexes of acquisitions (0, 1, 2, etc.) and the coefficient is defined in order to permit a display in seconds.

9.1.3.3 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	No card detected PCI-1711 with the set card number
3	Incorrect frequency by channel (must be between 1 and 50000 Hz)
4	Incorrect values range for a channel group

9.1.4 Card Advantech PCI-1715

Version 1.0



This driver permits the usage of the boards Advantech PCI-1715.

The card has 32 analog inputs 12 bits with programmable gain.

The driver provided by the builder for Windows can be installed beforehand in the goal to examine the good performance of the card with the provided utilities.

However, the installation of the Advantech driver is not obligatory to use the card with Optima PLC. The presence of this one as well as its base address are automatically determined by Optima PLC, that performs some direct accesses then to the hardware in order to permit the obtaining of real-time performances.

9.1.4.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) on the card permit to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

- Frequency by channel

Number of acquisitions wished per second and by channel. Attention, this frequency, in particular if it is raised, cannot always be gotten in an exact manner. The log file mentions to the start of the application an approximation of the frequency if this one was necessary.

The reading of the FIFO of the card consumes a relatively elevated CPU time, a global frequency of acquisition too high risks therefore to penalize the foreground functions of the application (user interface), or even to block them. A global frequency of 200000 acquisitions per second, seem a reasonable limit. It is possible to modulate the frequency of the acquisitions with the functions provided by the PCI1715 library, that permits not to penalize the rest of the system permanently if an elevated frequency is only useful to precise moments.

- Number of values in sample tables

Size of sample tables attached to every channels, permitting to perform background acquisitions (see PCI1715 library).

- Scale of measure channels x to y

Scale and type of measure for the group of 2 channels specified.

9.1.4.2 Usage

In the development workshop

The inputs/outputs must be declared either in a part "global Variables." The analog inputs are of LREAL type.

Example

```
Channel1 ANI0 at: LREAL; // first analog input
Channel2 ANI1 at: LREAL; // second analog input

ChanOut1 ANO16 at: LREAL; // first analog output
```

To use them in the program, it is possible to use the variable (ex: ANI0) or its mnemonic (ex: Channel1).

The values are returned in Volts, while taking into account the scale.

In the runtime Optima PLC

The inputs/outputs are displayable in the menu "PLC -> Variable -> Analog input ANI" and "Analog output ANO."

Usage of the sample tables

The functions of acquisitions in tables are available while using the PCI1715 library. Data are stocked in these tables to the asked acquisition frequency.

The tables are named of the following manner :

```
PCI1715_ < Card ID > _ < Nb of channel >
```

Thus, the channel 10 of the card 0 will be stocked in the PCI1715_0_10 table.

The stocked values are integer, between 0 and 4095 if the channel is unipolar, between -2048 and 2047 if the channel is bipolar. For every table, the display coefficient is defined according to the acquisition range of the channel, what permits a display in V.

Besides, the "time" is stocked in a table :

```
PCI1715_ < Card ID > _Time
```

The stored values are the indexes of acquisitions (0, 1, 2, etc.) and the coefficient is defined in order to permit a display in seconds.

9.1.4.3 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	No card detected PCI-1710 with the set card number

3	Incorrect frequency by channel (must be between 1 and 50000 Hz)
4	Incorrect values range for a channel group

9.1.5 Card Advantech PCI-1723

Version 1.0



This driver permits to use of the card Advantech PCI-1723.

The card has 8 analog outputs 16 bits, and of 16 digital channels of entry or exit.

The driver provided by the builder for Windows can be installed beforehand in the goal to examine the good performance of the card with the utilities provided by the builder.

However, the installation of the Advantech driver is not obligatory to use the card with Optima PLC. The presence of this one as well as its base address are automatically determined by Optima PLC, that performs some direct accesses then to the hardware in order to permit the obtaining of real-time performances.

The analog output channels are actuated in a synchronous manner.

The 16 digital inputs/outputs can be programmed in input or in output by group of 8.

9.1.5.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (S1) of the card permits to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

- Way of the digital inputs/outputs 0 to 7

The first 8 digital I/O can be programmed either as input or as output.

- Way of the digital inputs/outputs 8 to 15

The last 8 digital I/O can be programmed either as input or as output.

9.1.5.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	No card detected PCI-1723 with the set card number

9.1.6 Card Advantech PCI-1733

Version 1.2



This driver provides access to the 32 inputs of the card Advantech PCI-1733.

The driver provided by the manufacturer for Windows can be installed in advance in order to test the proper functioning of the card with the provided utilities.

However, the installation of driver Advantech is not required to use the card with Optima PLC. The presence thereof as well as its base address is automatically determined by Optima PLC, which then direct access to equipment to enable the achievement of real-time performance.

9.1.6.1 Parameters

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) on the card permit to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

9.1.6.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	Undetected card with the given number

9.1.7 Card Advantech PCI-1734

Version 1.2



This driver provides access to the 32 outputs of the card Advantech PCI-1734.

The driver provided by the manufacturer for Windows can be installed in advance in order to test the proper functioning of the card with the provided utilities.

However, the installation of driver Advantech is not required to use the card with Optima PLC. The presence thereof as well as its base address is automatically determined by Optima PLC, which then direct access to equipment to enable the achievement of real-time performance.

9.1.7.1 Parameters

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) on the card permit to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

9.1.7.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	Undetected card with the given number

9.1.8 Card Advantech PCI-1747U

Version 1.0



This driver gives access to the analog inputs of the card Advantech PCI-1747U.

The card has 64 analog inputs 16 bits with programmable gain.

The driver provided by the builder for Windows can be installed beforehand in the goal to examine the good performance of the card with the utilities provided by the builder.

However, the installation of the Advantech driver is not obligatory to use the card with Optima PLC. The presence of this one as well as its base address are automatically determined by Optima PLC, that performs some direct accesses to the hardware in order to permit the obtaining of real-time performances.

The driver can be used jointly to the PCI1747U library, that permits a storage of the values acquired in tables.

9.1.8.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) of the card permits to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

- Frequency by channel

Number of acquisitions wished per second and by channel. Attention, this frequency, in particular if it is raised, cannot be gotten in an exact manner. The log file mentions to the start of the application an approximation of the

frequency if this one was necessary.

The reading of the FIFO of the card consumes a relatively elevated CPU time, a global frequency of acquisition too high risks therefore to penalize the foreground functions of the application (user interface), or even to block them. A global frequency of 100000 acquisitions per second, all disconcerted ways, seems a reasonable limit. It is possible to modulate the frequency of the acquisitions with the functions provided by the PCI1747U library, what permits not to penalize the rest of the system permanently if an elevated frequency is only useful to precise moments.

- Number of values in tables

Size of the samples tables attached to every channel, permitting to perform background acquisitions task (see PCI1747U library).

- Scale of measure channels x to y

Scale and type of measure for the group of 4 channels specified.

9.1.8.2 Usage

In the development workshop

The inputs/outputs must be declared either in a part "global Variables." The analog inputs are of LREAL type.

Example

```
Channel1 ANI0 at: LREAL;
Channel2 ANI1 at: LREAL;
```

To use them in the program, it is possible to use the variable (ex: ANI0) or his/her/its mnemonic (ex: Channel1).

The values are returned in Volts, while taking into account the scale.

In the runtime Optima PLC

The inputs/outputs are displayable in the menu "PLC - > Variable - > Analog input ANI."

Usage of the sample tables

The functions of acquisitions in tables are available by using the PCI1747U library. Data are stocked in these tables to the asked acquisition frequency.

The tables are named of the following manner :

```
PCI1747U_ < Card ID > _ < Nb of channel >
```

Thus, the channel 10 of the card 0 will be stocked in the PCI1747U_0_10 table.

The stocked values are integer, between 0 and 65535 if the channel is unipolar, between -32768 and 32767 if the way is bipolar. For every table, the coefficient of display is defined according to the acquisition range of the channel, what permits a display in V.

Besides, the "time" is stocked in a table :

```
PCI1747U_ < Card ID > _Time
```

The stocked values are the indexes of acquisitions (0, 1, 2, etc.) and the coefficient is defined in order to permit a display in seconds.

9.1.8.3 Status values

Statute	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance

1	Bad card number (must be understood between 0 and 15)
2	No card detected PCI-1747U with the set card number
3	Frequency by incorrect channel (must be between 1 and 50000 Hz)
4	Incorrect values range for a channel group

9.1.9 Card Advantech PCI-1751

Version 1.2



This driver provides access to inputs/outputs of the card Advantech PCI-1751.

The driver provided by the manufacturer for Windows can be installed in advance in order to test the proper functioning of the card with the provided utilities.

However, the installation of driver Advantech is not required to use the card with Optima PLC. The presence thereof as well as its base address is automatically determined by Optima PLC, which then direct access to equipment to enable the achievement of real-time performance.

9.1.9.1 Parameters

- Number of the card

The system can contain several card PCI 1751. This parameter permits to specify which is addressed here. The number of the first card is 1. This is the default value.

- Mask of inversion IO 0 to 7
- Mask of inversion IO 8 to 15
- Mask of inversion IO 16 to 23
- Mask of inversion IO 24 to 31
- Mask of inversion IO 32 to 39

These settings help define a reversal for each IO compared to the state reported by the card. It is a mask which bits to 1 indicate a reversal of the corresponding.

For example, the value of \$21 mask for the mask inversion IO 0 to 7, said that the IO 0 and 5 driver must be reversed.

The defaults are \$FF, because the original state of each entry is 1 in the absence of signal

9.1.9.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 1 and 16)
2	Undetected card

9.1.10 Card Advantech PCI-1762

Version 1.2



This driver provides access to 16 isolated inputs and 16 relay outputs of the card Advantech PCI-1762. 16 supplementary inputs for control of the output relay states are also available.

The channel numbers are used as follows :

First channel	Last channel	use
0	15	Isolated inputs of the card
16	31	Inputs for relays state control
32	47	Output relays of the card

The driver provided by the manufacturer for Windows can be installed in advance in order to test the proper functioning of the card with the provided utilities.

However, the installation of driver Advantech is not required to use the card with Optima PLC. The presence thereof as well as its base address is automatically determined by Optima PLC, which then direct access to equipment to enable the achievement of real-time performance.

9.1.10.1 Parameters

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) on the card permit to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

9.1.10.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 1 and 16)
2	Undetected card with given number.

9.1.11 Card Advantech PCI-1780U

Version 1.0



This card is mainly used for its 16-bit 8 counters. However, the driver can also take advantage of 8 digital inputs and 8 digital outputs it entails.

The driver provided by the manufacturer for Windows can be installed in advance in order to test the proper functioning of the card with the provided utilities.

However, the installation of Advantech driver is not required to use the card with Optima PLC. The presence thereof as well as its base address is automatically determined by Optima PLC, which then direct access to equipment to enable the achievement of real-time performance.

The driver can be used in conjunction with the library PCI1780U, which enables the programming of counters.

The first 8 digital channels of the card (0 to 7) are inputs, 8 following (8 to 15) are outputs.

9.1.11.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

- Number of the card

This parameter allows Optima PLC to identify the card possibly among several present. 4 switches (SW1) of the card permit to define an identification number of 0 to 15 (0 by default). This number must be indicated here.

9.1.11.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 0 and 15)
2	No card detected PCI-1780U with the set card number

9.1.12 Modules Adam

9.1.12.1 Adam 4017, 4018

Version 1.1



These modules provide 8 analog 16 bits inputs on bus 485.

Communication with a module is done via a serial communication port RS485, which must be configured in the configuration of serial ports Optima PLC.

The module configuration (baud rate, address, etc..) and its possible calibration can be performed with the software provided by Advantech. Driver operation in Optima PLC assumes that this configuration is already done and that access to the module is working correctly.

9.1.12.1.1 Parameters

To configure the driver, select it from the list of drivers and click the "Settings" button.

- Number of serial port

Enter the number of COM port used for communication with the module. This COM port must also be configured and enabled in Optima PLC.

- Module address

This parameter corresponds to the address given to the module with the software Advantech Adam.

- Communication speed

Enter the data communication speed when configuring the module. This speed must also be consistent with that COM port.

- Display error message

If "yes" is selected, a failure to communicate with the module (except during application initialization, where another type of message is consistently displayed) is indicated on the screen.

9.1.12.1.2 Status values

Status	Description
-1	Driver not initialized
-2	Driver stopped in the stopping phase of the application
0	Nominal performance
1, 241	Communication error (no answer)
2, 242	Failure to communicate
3, 243	COM port not active
4, 244	Module address incorrect
5, 245	A single module expected on the COM port (deprecated)
6, 246	Module address duplicated on the COM port
7, 247	Inconsistency of speed used on the COM port

255

Module not used

9.1.12.2 Adam 4050, 4053, 4055

Version 1.1



These modules provide inputs / outputs on RS485 bus.

	Inputs	Outputs
Adam 4050	7 (channels 0 à 6)	8 (channels 8 à 15)
Adam 4053	16 (channels 0 à 15)	0
Adam 4055	8 (channels 0 à 7)	8 (channels 8 à 15)

Communication with a module is done via a serial communication port RS 485, which must be configured in the configuration of serial ports Optima PLC.

The module configuration (baud rate, address, etc..) and its possible calibration can be performed with the software provided by Advantech. Driver operation in Optima PLC assumes that this configuration is already done and that access to the module is working correctly.

9.1.12.2.1 Parameters

To configure the driver, select it from the list of drivers and click the "Settings" button.

- Number of serial port

Enter the number of COM port used for communication with the module. This COM port must also be configured and enabled in Optima PLC.

- Module address

This parameter corresponds to the address given to the module with the software Advantech Adam.

- Measuring range

This parameter sets the full scale and type of measured signals.

- Communication speed

Enter the data communication speed when configuring the module. This speed must also be consistent with that COM port.

- Display error message

If "yes" is selected, a failure to communicate with the module (except during application initialization, where another type of message is shown) is indicated on the screen.

- compatibility mode

This parameter is used to simplify porting Apigraf applications to Optima PLC. In normal mode, the value of channels available in the application is exactly the floating point value returned by the module. In compatibility mode Apigraf, the value is multiplied by a factor depending on the scale chosen so that there is no fractional part (the analog channels being sent back to integer values in Apigraf).

9.1.12.2.2 Status values

Status	Description
-1	Driver not initialized
-2	Driver stopped in the stopping phase of the application
0	Nominal performance
1, 241	Communication error (no answer)
2, 242	Failure to communicate
3, 243	COM port not active
4, 244	Module address incorrect
5, 245	A single module expected on the COM port (deprecated)
6, 246	Module address duplicated on the COM port
7, 247	Inconsistency of speed used on the COM port
255	Module not used

9.2 Hilscher

9.2.1 Hilscher generic CIF board

Version 1.0



This driver allows management by Optima PLC of an Hilscher CIF generic card.

The principle of use is as follows: initially, the card is configured and tested with SyCon (configuration utility from Hilscher). The configuration thus obtained is postponed then used in Optima PLC in IO mode (ie reading and writing directly in the DPM (Dual Port Memory) of the card).

9.2.1.1 Installation

The following steps must be followed to get a correct performance of the Hilscher card with Optima PLC.

- Installation of the Hilscher driver
- Installation of the Hilscher software of configuration SyCon
- Definition of a configuration with SyCon and uploading on the card. The mode of handshake must be "uncontrolled" with this version of the driver.
- Test of the configuration with SyCon (reading of the inputs, writing of the outputs)

Important : Optima PLC is not able to operate the card if the performance is not already correct with SyCon.

The [steps of the parameterization](#) ^[403] are retailed in the following paragraphs.

9.2.1.2 Parameters

Parameters to enter are :

- Number of the card

Up to four Hilscher cards can be installed. Specify in this parameter the number of the present card (1 to 4).

- Activation of the Watch Dog

If the watch dog is activated, the card software automatically cuts the outputs in case of failure of the PC. Specify Yes or No.

- Automatic restart after defect

According to the parameterization performed in SyCon, the card cuts the communication at the time of an incident on the network. If this parameter is indexed on Yes, Optima PLC automatically provokes a reset of the card and therefore a restart of the communication if the incident is repaired.

If the parameter is indexed on No, the communication is not automatically restarted. A reset of the card must be performed. This reset can be made by different means :

- Restart of the application (a reset is performed to the start).
- Reset performed in SyCon (it is possible to start this software during execution of the application Optima PLC).
- Use the Hilscher_CIF library and the CIF_Reset function or CIF_Reinit (call one of these functions on a condition set for example from a page of supervision).

9.2.1.3 Steps of the parameterization

Here is an example showing how to quickly parameterize a network with a Hilscher card and Optima PLC. The network used here is Interbus, but the process is very near for the other networks (Profibus, DeviceNet, etc...).

First of all, use the CD provided by Hilscher to install the "Device driver" and the utilitarian "SyCon." Server OPC is not required by Optima PLC. Then install the card physically in the PC.

In the Windows "start menu", call the utility of driver configuration to verify the detection of the card.



The following window lists the detected cards.



Configuration of SyCon

In the "start menu", call the SyCon software.

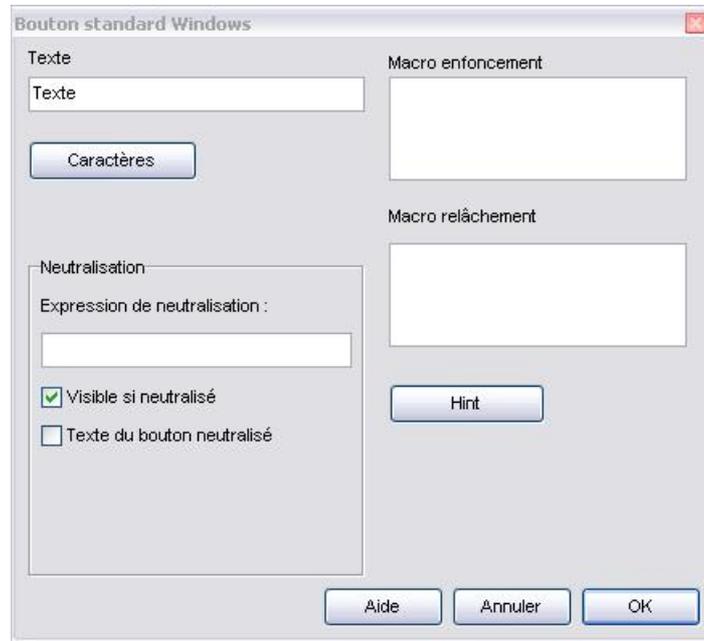


The basis window of the software is displayed. The project is initially empty. Click on the following icon to insert a

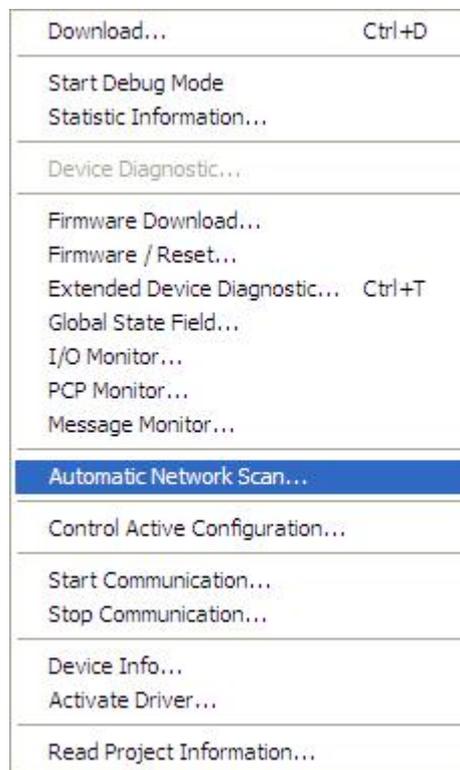
"master" on the network :



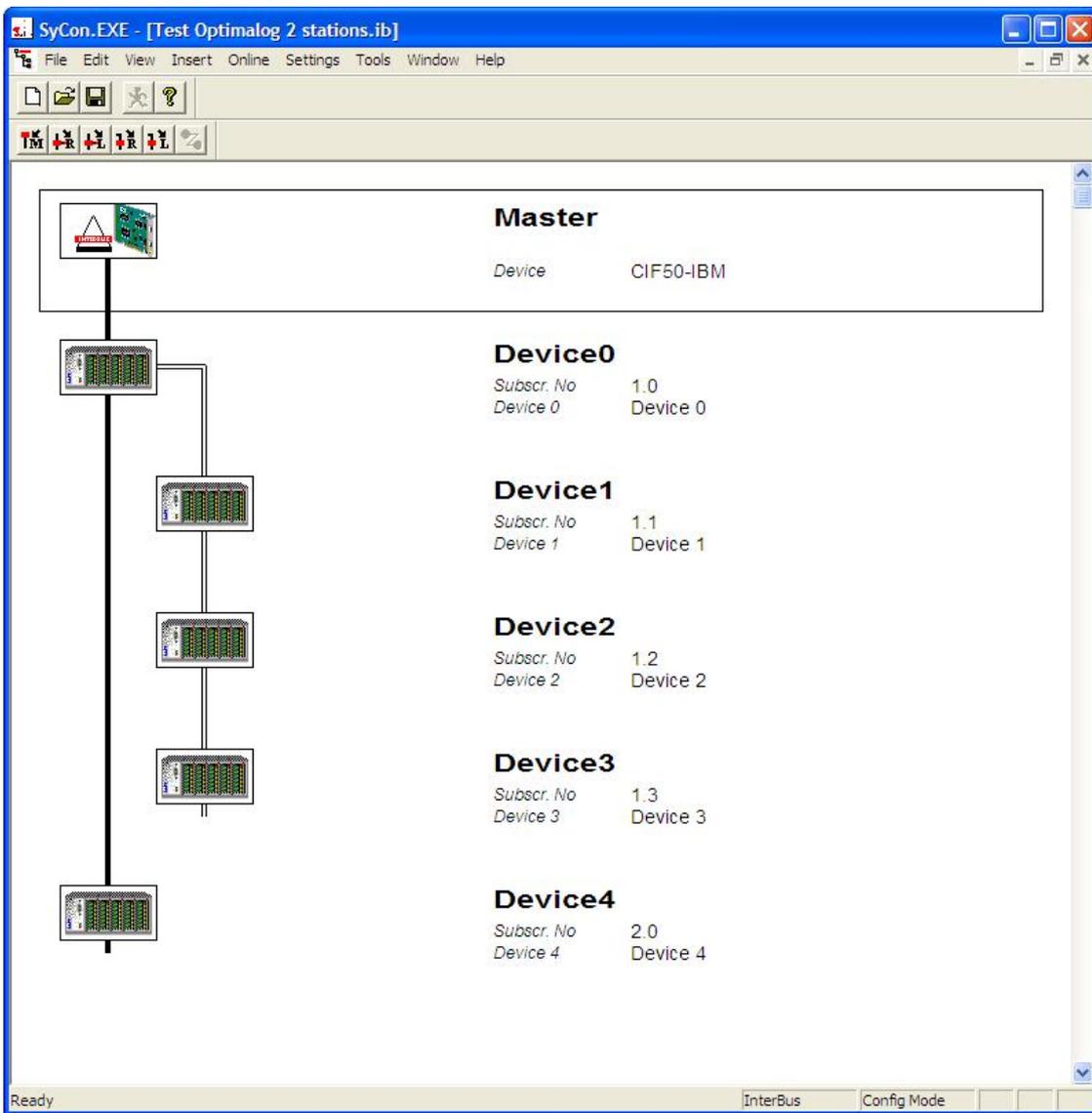
Choose in the list the card type to use. In the example it is a CIF50-IBM card dedicated to the Interbus protocol.



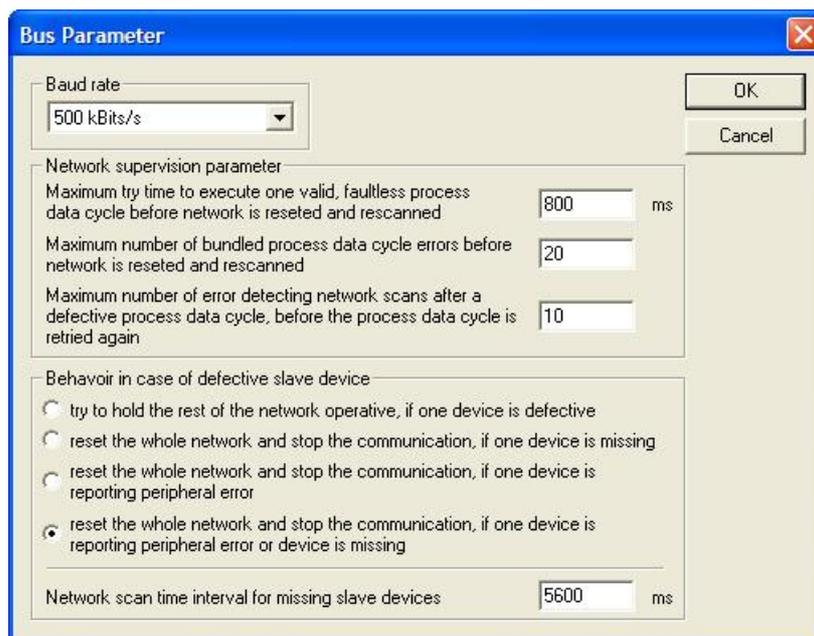
In the menu "Online", ask for an automatic scan of the network, in order to detect the present configuration.



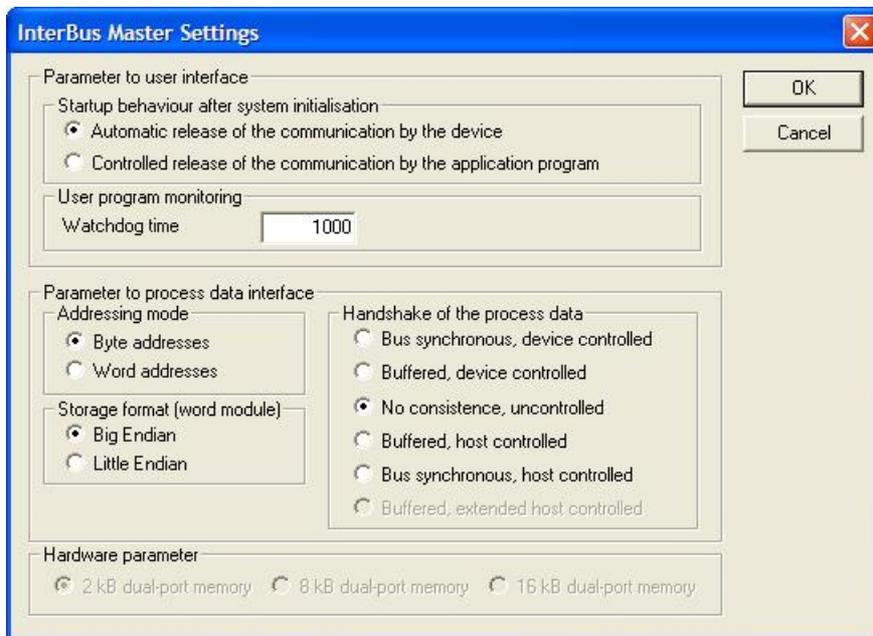
The scan must be performed with the CIF card...



In the menu "Settings", parameterize the communication and the master. The configuration of the fieldbus depends on this one.



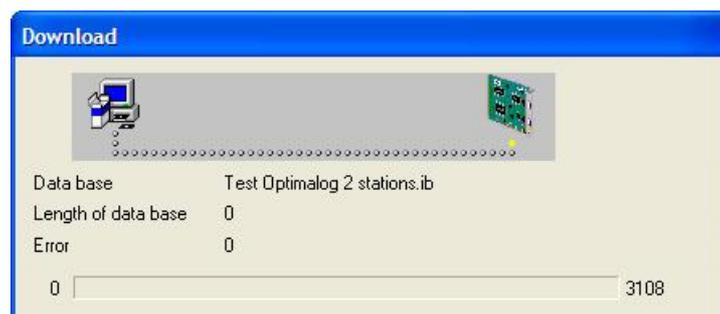
Configuration of the master (Hilscher card). The mode "no consistency, uncontrolled" is imperative with the present version of the driver. It means that there is not a synchronization in the release of the information in input or in output (to see Hilscher documentation).



When the configuration is finished, this one must be uploaded on the card ("Online" menu).



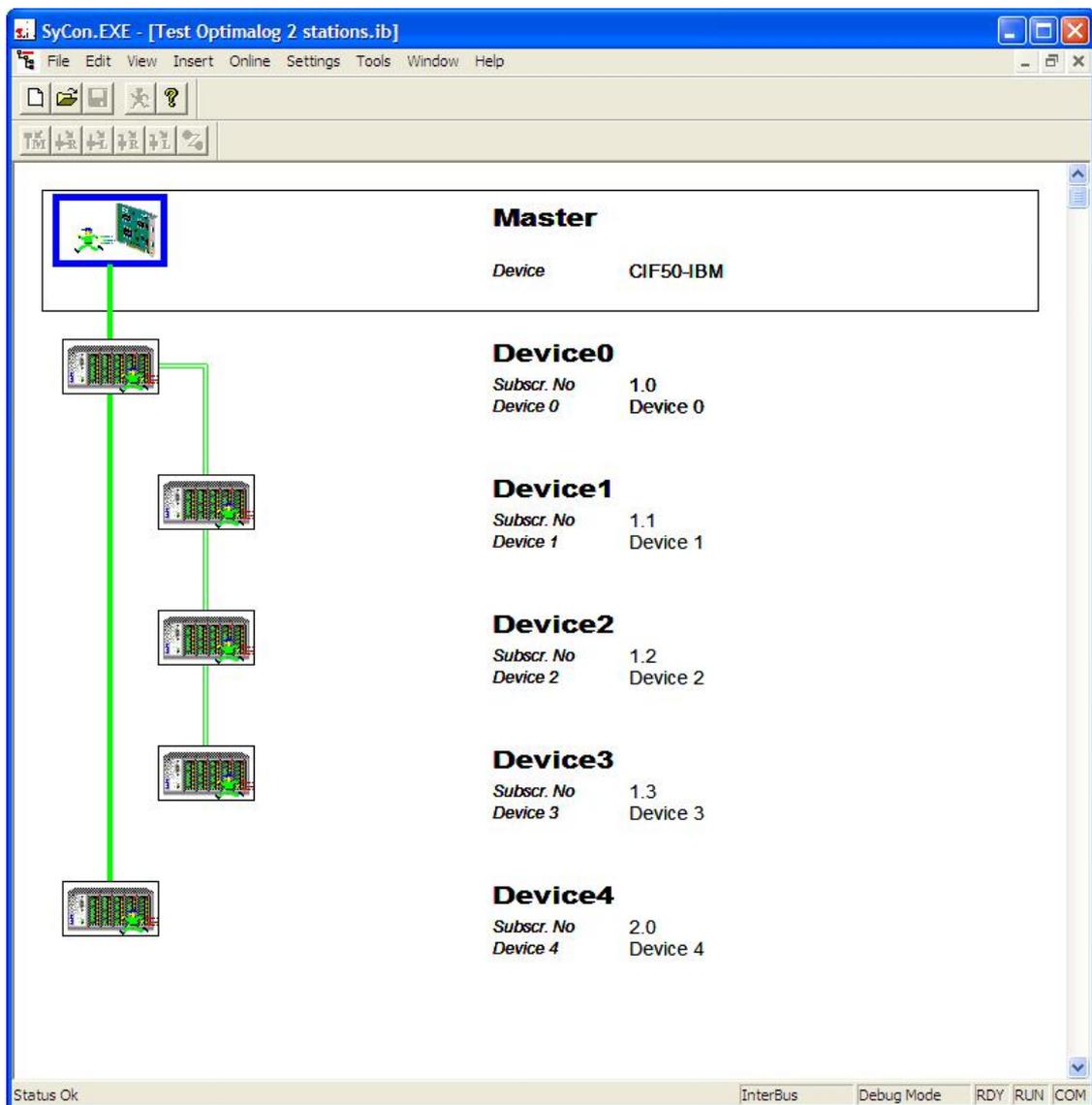
The code of configuration is sent then to the card, that stocks it in flash memory. This configuration is then permanent.



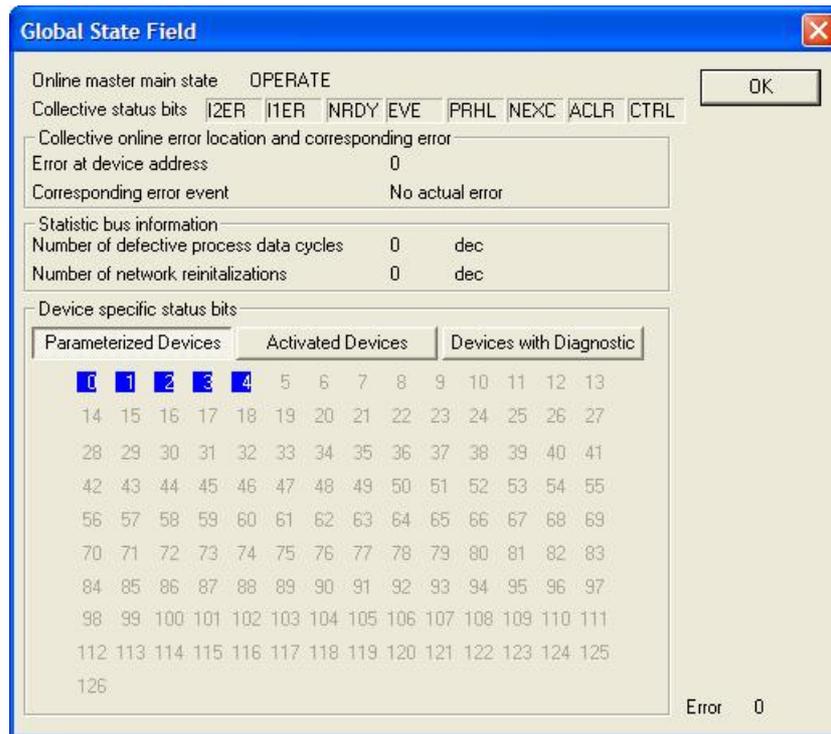
Once the uploading finished, if the "Automatic parameter release of the communication by the device" has been selected, the card immediately begins to interact. Otherwise, the dialog can be started by hand ("Online" menu).



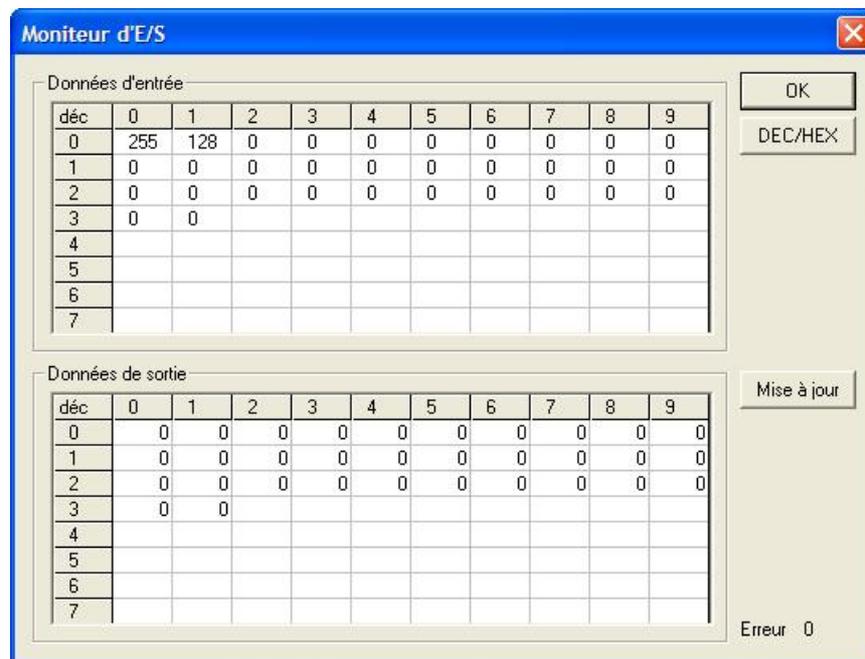
The statement of the network can also be examined in "Debug Mode" in the menu "Online", here is the result of it :



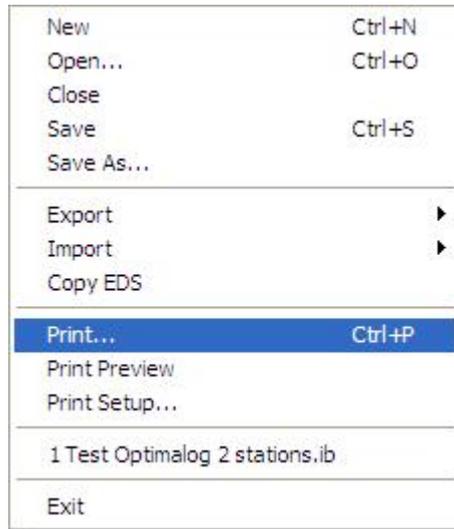
When the communication is started (out "Debug Mode"), information of diagnosis are available and permit to audit the performance of the facilities and to discover the source of a breakdown. Below, the display "Global state fields" indicates the general statement of the card and the managed facilities (set, detected and in defect).



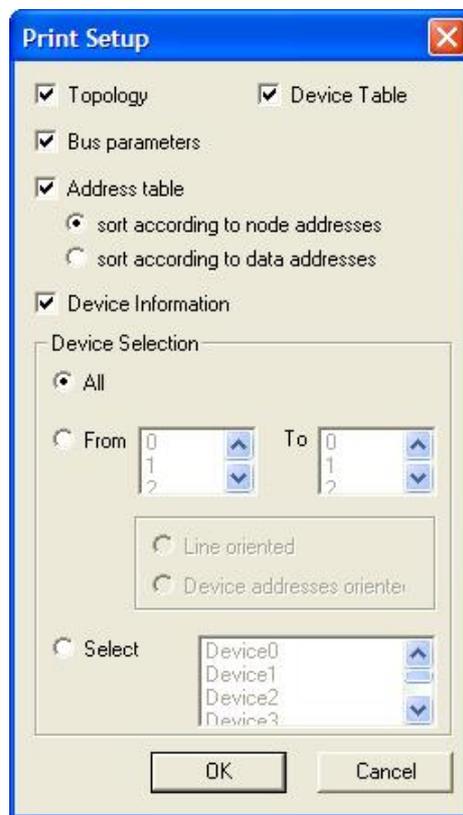
The option "IO monitor" permits to examine the state of the inputs and outputs. The zones of inputs and outputs are displayed as bytes, the outputs can be forced.



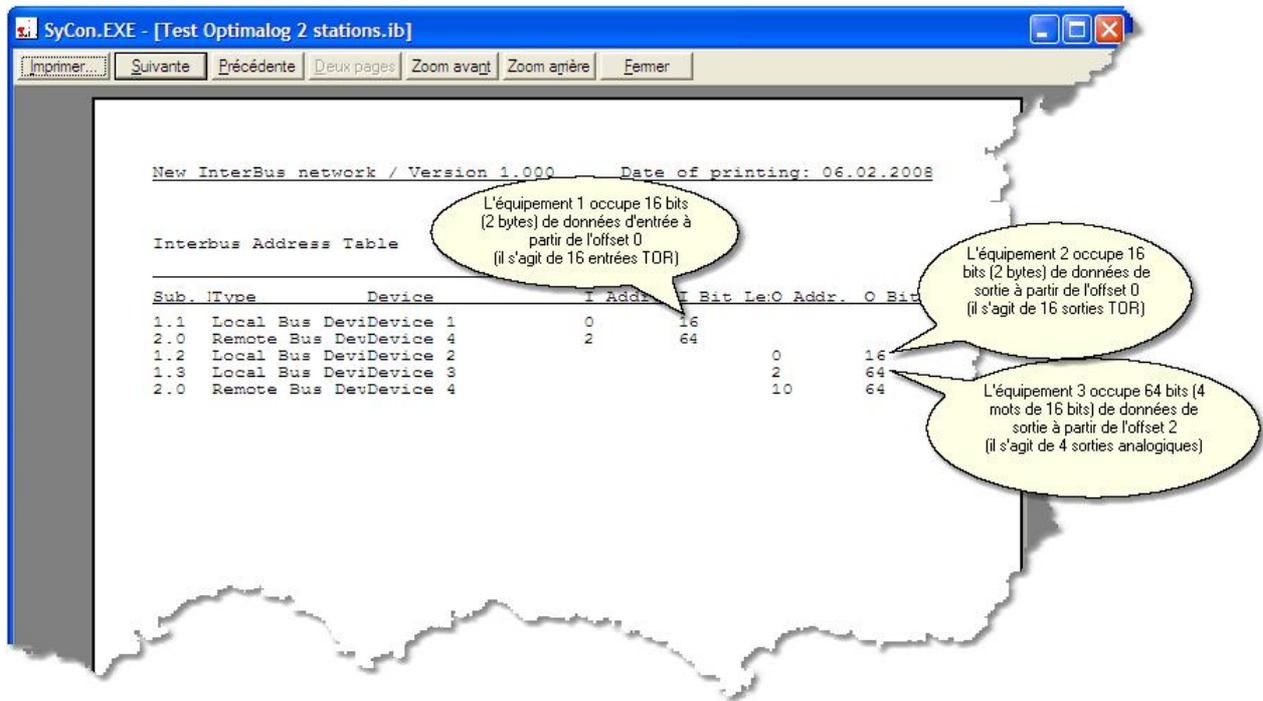
When the performance is OK, it is possible to pass to the parameterization of Optima PLC. In this goal, it is necessary to get the synthesis of the settings of every fieldbus equipment in the input and output areas of the card. For it, the simplest is to print the report of the project :



The following window permits to select the information to print in the report. For the configuration with Optima PLC, it is the "Address table" that is interesting here :

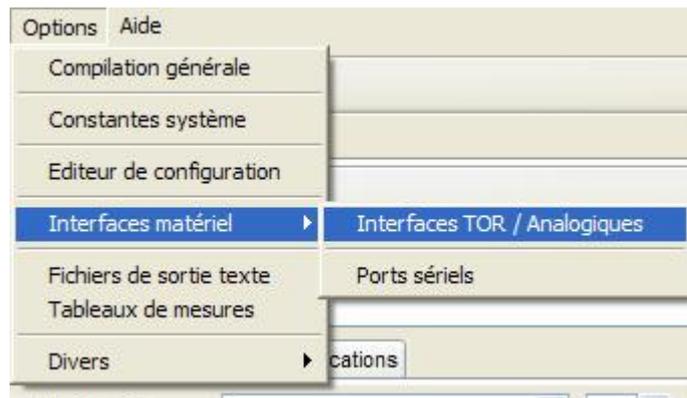


Here is the répartition of data gotten for the configuration of our Interbus network. For every device, the size of data in input or output, as well as their position in the tables of the card, is indicated. The equipment 0 doesn't appear here, because it doesn't comprise any input/output (it is here about a "head of station").



Configuration of optima PLC

In the workshop, call the configuration of the interfaces.



In the list of the interfaces configured, it is necessary to define an interface for the CIF card, (without input/output directly available) and an interface for every equipment of the network (here the modules 1 to 4).

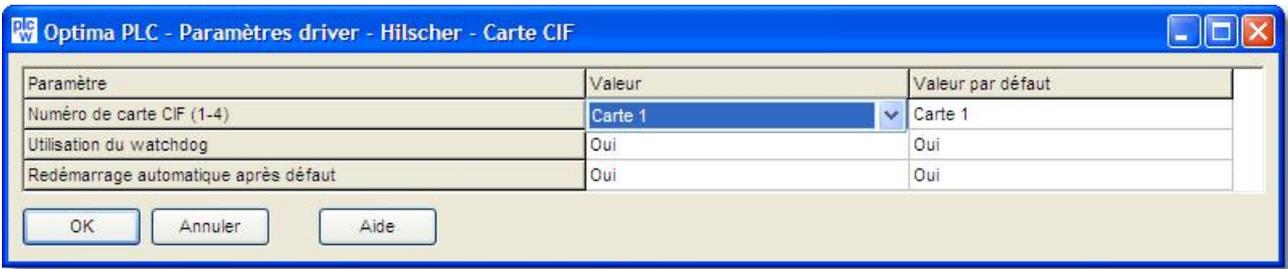
Optima PLC - Drivers configurés

Index	Libellé	TOR début	TOR fin	ANA début	ANA fin	Simulation
1	Bus PCI - Advantech - PCI-1747U			0	63	
2	Bus PCI - Advantech - PCI-1780U	0	15			
3	Bus PCI - Advantech - PCI-1723	16	31	64	71	
4	Hilscher - Carte CIF					
5	Hilscher - Station déportée - Générique	32	47			
6	Hilscher - Station déportée - Générique	48	63			
7	Hilscher - Station déportée - Générique			72	75	
8	Hilscher - Station déportée - Générique			76	83	

Ajouter Supprimer Paramètres Aide

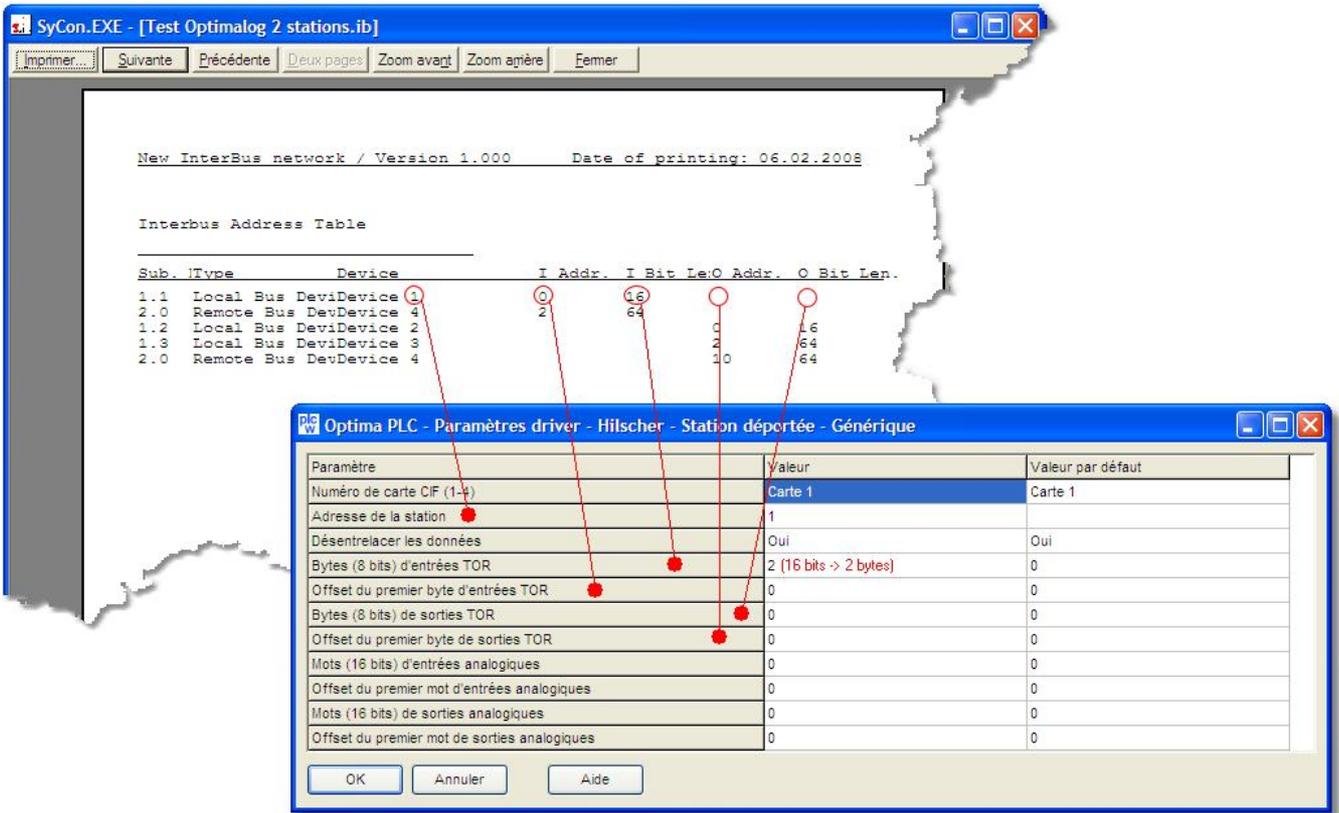
OK Annuler Infos

Configuration of the CIF card :



Devices configuration.

For that step, use the SyCon configuration report, especially the address table. The parameters can be then immediately transposed as shown below :



Device configuration 1 :



Device configuration 2 :

Paramètre	Valeur	Valeur par défaut
Numéro de carte CIF (1-4)	Carte 1	Carte 1
Adresse de la station	2	
Désentrelacer les données	Oui	Oui
Bytes (8 bits) d'entrées TOR	0	0
Offset du premier byte d'entrées TOR	0	0
Bytes (8 bits) de sorties TOR	2	0
Offset du premier byte de sorties TOR	0	0
Mots (16 bits) d'entrées analogiques	0	0
Offset du premier mot d'entrées analogiques	0	0
Mots (16 bits) de sorties analogiques	0	0
Offset du premier mot de sorties analogiques	0	0

Buttons: OK, Annuler, Aide

Device configuration 3 :

Paramètre	Valeur	Valeur par défaut
Numéro de carte CIF (1-4)	Carte 1	Carte 1
Adresse de la station	3	
Désentrelacer les données	Oui	Oui
Bytes (8 bits) d'entrées TOR	0	0
Offset du premier byte d'entrées TOR	0	0
Bytes (8 bits) de sorties TOR	0	0
Offset du premier byte de sorties TOR	0	0
Mots (16 bits) d'entrées analogiques	0	0
Offset du premier mot d'entrées analogiques	0	0
Mots (16 bits) de sorties analogiques	4	0
Offset du premier mot de sorties analogiques	2	0

Buttons: OK, Annuler, Aide

Device configuration 4 :

Paramètre	Valeur	Valeur par défaut
Numéro de carte CIF (1-4)	Carte 1	Carte 1
Adresse de la station	4	
Désentrelacer les données	Oui	Oui
Bytes (8 bits) d'entrées TOR	0	0
Offset du premier byte d'entrées TOR	0	0
Bytes (8 bits) de sorties TOR	0	0
Offset du premier byte de sorties TOR	0	0
Mots (16 bits) d'entrées analogiques	4	0
Offset du premier mot d'entrées analogiques	2	0
Mots (16 bits) de sorties analogiques	4	0
Offset du premier mot de sorties analogiques	10	0

Buttons: OK, Annuler, Aide

9.2.1.4 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance

1	Bad card number (must be between 1 and 4)
2	Board already used
3	Key of Windows register incorrect concerning this board (driver not installed correctly)
4	Incorrect address in the Windows register
5	Incorrect board size in the Windows register
6	Internal defect of data definition in the driver Optima PLC
7	Non correctly identified board
8	Allocation of a zone of access to the card impossible
9	Mode of handshake incorrect (must be Uncontrolled).
10	Non ready board (Information Ready to 0).
11	Board not in run (Information Run to 0).
12	Board not in communication (Information COM to 0).

9.2.1.5 Stations

9.2.1.5.1 Hilscher - generic Station deportees

Version 1.0

This driver is conceived for an usage bound to the one of the [Hilscher generic CIF board](#)^[402]. He/it permits modéliser the data access of entry and exit of a station via the corresponding data extents of the Hilscher card.

9.2.1.5.1.1 Parameters

- Number of the CIF card

Number of the card to which the station is connected

- Address of the station

Number of address of the station on the network

- Désentrelacer data

This option permits to palliate the problem bound to the order of the bytes of data. If the order is: very consistent weight of least significant, ("big endian") data must be désentrelacées. The normal mode of the PC is: consistent least significant of strong weight ("little endian").

The following parameters must be enter according to the apportionment of data of the station in the zones memorials of the Hilscher card. This apportionment can be gotten in the SyCon middleware, once the correctly configured network.

- Bytes of TOR inputs

Number of bytes of data of TOR (accessible bits to bits in Optima PLC) inputs.

- Offset of the first byte of TOR inputs

Address of the first byte of TOR inputs in the input area of the Hilscher card.

- Bytes of TOR outputs

Number of bytes of data of TOR (accessible bits to bits in Optima PLC) outputs.

- Offset of the first byte of TOR outputs

Address of the first byte of TOR outputs in the input area of the Hilscher card.

- Words of analog inputs

Number of words of 16 bits of data of inputs (accessible under numeric shape in Optima PLC).

- Offset of the first word of analog inputs

Address of the first word of analog inputs in the input area of the Hilscher card.

- Words of analog outputs

Number of words of 16 bits of data of outputs (accessible under numeric shape in Optima PLC).

- Offset of the first word of analog outputs

Address of the first word of analog outputs in the input area of the Hilscher card.

9.2.1.5.1.2 Securities of statute

Statute	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be understood between 1 and 4)
2	Non definite card
3	Incoherence of definition of the input areas or exit
100	Non definite station in the project
101	Definite station in the project but undetected
102	Station in defect non communiqué by the card

9.2.1.5.2 Hilscher - Station deportees 16l 16O

Version 1.0

This driver is identical to the generic station, but limits itself to the management of 16 bits of TOR inputs and 16 bits of TOR outputs.

9.2.2 Card Hilscher CIF CAN Layer 2

Version 1.0



This driver permits the management by Optima PLC of a card Hilscher CIF CanOpen used in transparent text mode (Layer 2). In this mode, the application is in charge of the direct management of the CAN messages received and messengers, who are not managed therefore by the card herself as it is the case with the CanOpen protocol or DeviceNet.

The installation and the usage of the card make themselves of manner similar to the case of the [card generic Hilscher](#)^[402]. The values of statute are also identical.

The mailing and the receipt of the CAN messages are performed via the library Hilscher_CIF and the functions dedicated to the CAN Layer 2.

9.2.3 Card Hilscher CIF 50-in Modbus / TCP IO mode (server)

This driver permits to use the card Hilscher CIF in mode server, that is to say that the card constitutes a Modbus / TCP "slave" answering the solicitations of Modbus / TCP masters of the network.

The driver comprises 2048 TOR inputs and 2048 TOR outputs. These inputs/outputs correspond to the variables that the main Modbuses / TCP can come to read and to write the outside.

In the same way 256 channels 16 bits in inputs and 256 channels 16 bits in outputs are available, to the same locations that the inputs/outputs TOR.

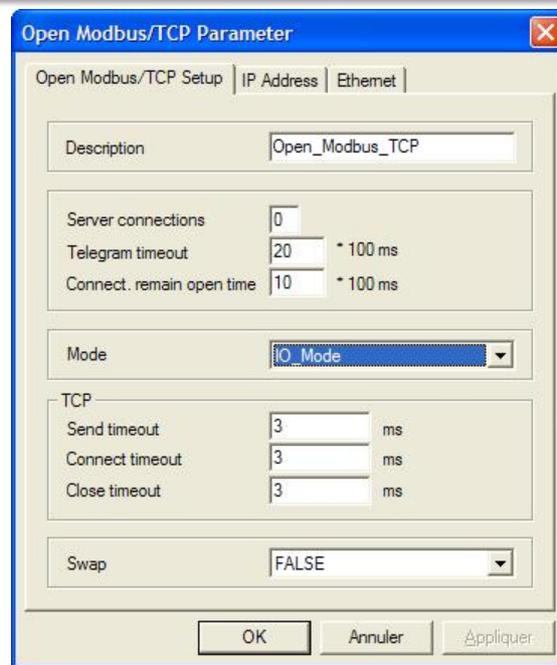
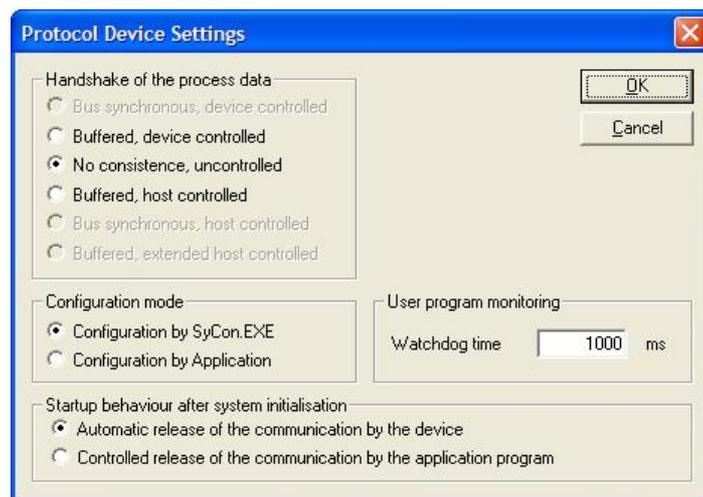
Distribution of the variables :

Entered TOR	indexes 0 to 2047
Taken TOR	indexes 2048 to 4095
Analog inputs	indexes 0 to 255
Analog outputs	indexes 256 to 511

9.2.3.1 Installation

The following steps must be followed to get a performance correct of the Hilscher card with Optima PLC.

- Installation of the Hilscher driver
- Installation of the software Hilscher of SyCon configuration
- Definition of a configuration with the help of SyCon and downloading on the card. The mode of handshake used must be "uncontrolled" with this version of the driver. The mode Modbus / TCP performance must be "IO mode" and non Message mode."



- Test of the configuration with SyCon (reading of the inputs, entry of the outputs)

Importing: PLC optima are not able to operate the card if the performance is not already correct with SyCon.

9.2.3.2 Parameters

Parameter to enter coins Optima PLC :

- Number of the card

Until four Hilscher cards can be installed. To specify in this parameter the number of the present card (1 to 4).

9.2.3.3 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad CIF card number (must be between 1 and 4)
2	Board already definite
3	Hilscher driver non installed
4	Incorrect board address in the register
5	Memory size of the board incorrect
6	Version of the driver Optima PLC incorrect
7	Incorrect board identifier
8	Impossible to allocate the memory zone of the board
9	Mode of Handshake configured badly on the board (the mode must be Uncontrolled, to reconfigure with SyCon)
10	Incorrect board type
11	Incorrect board model
12	Mode of Modbus management incorrect (must be configured with SyCon to IO mode)
20	Non ready board
21	Board not in Run mode
22	Board not in statement to communicate

9.2.4 Card Hilscher CIF 50-En Modbus/TCP message mode (client)

This driver permits to use the card Hilscher CIF in client mode, that is to say that the card is a main Modbus / TCP.

This driver doesn't place at entered no disposal / exit. His/her/its mode of performance is the next one: when the option is proposed, the Hilscher card is used by the Modbus functions for and on behalf of the Ethernet interface standard of the PC.

This option is proposed for example in the parameterization of the generic Modbus / TCP interface. To use the Hilscher card with this driver instead the interface standard Ethernet, prenez care to define the Hilscher driver before the generic Modbus / TCP driver in the list of the drivers, and in this last to specify the number of the Hilscher card to use, of 1 to 4.

You can also use in programming the functions usual of the Modbus library, in relation with a Modbus channel gotten by the ModbusGetHilscherChannel function.

Attention, the Hilscher card doesn't permit to use all functions usually provided by Optima PLC. The functions following Modbuses are only available: 1 to 7, 15 and 16.

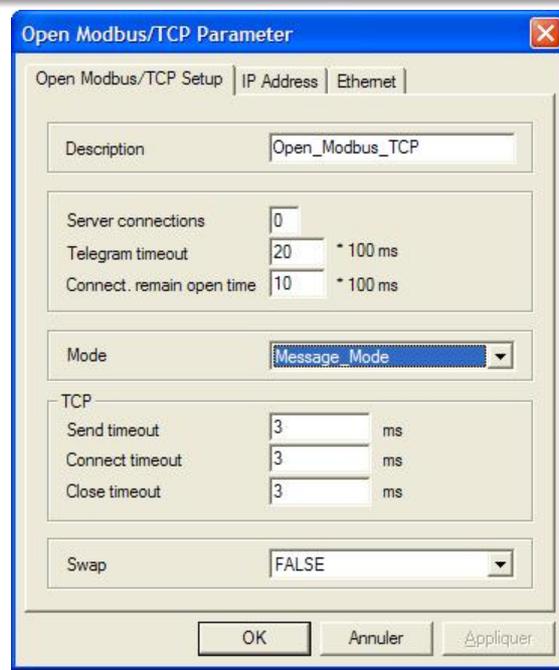
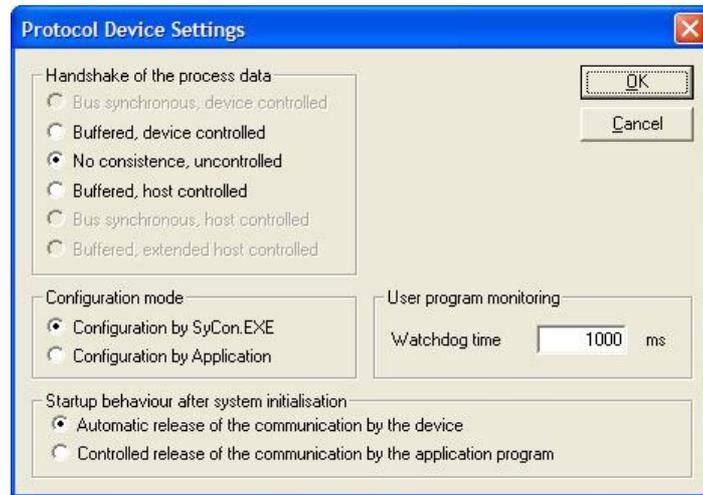
In this mode of performance, the card cannot be used as slave Modbus (server).

9.2.4.1 Installation

The following steps must be followed to get a performance correct of the Hilscher card with Optima PLC.

- Installation of the Hilscher driver

- Installation of the software Hilscher of SyCon configuration
- Definition of a configuration with the help of SyCon and downloading on the card. The mode of handshake used must be "uncontrolled" with this version of the driver. The mode Modbus / TCP performance must be "Message mode" and non IO mode."



- Test of the configuration with SyCon (broadcast / receipt of messages modbus).

Important : PLC optima are not able to operate the card if the performance is not already correct with SyCon.

9.2.4.2 Parameters

Parameter to enter coins Optima PLC :

- Number of the card

Until four Hilscher cards can be installed. To specify in this parameter the number of the present card (1 to 4).

9.2.4.3 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application

0	Nominal performance
1	Bad CIF card number (must be between 1 and 4)
2	Board already definite
3	Hilscher driver non installed
4	Incorrect board address in the register
5	Memory size of the board incorrect
6	Version of the driver Optima PLC incorrect
7	Incorrect board identifier
8	Impossible to allocate the memory zone of the board
9	Mode of Handshake configured badly on the board (the mode must be Uncontrolled, to reconfigure with SyCon)
10	Incorrect board type
11	Incorrect board model
12	Mode of Modbus management incorrect (must be configured with SyCon to IO mode)
20	Non ready board
21	Board not in Run mode
22	Board not in statement to communicate

9.3 Heidenhain

9.3.1 IK220 card

Version 1.0



This driver gives access at the ways of numbering of the card Heidenhain IK220.

This driver doesn't permit a real-time demand to data of the card, that is piloted via the driver provided by the builder. Besides, this particular structure explains that the driver can manage until 16 ways: indeed, until 8 cards propertied IK220 each 2 ways can be used simultaneously. This driver is therefore to be defined only once in the Optima PLC configuration in order to use one to eight cards.

9.3.1.1 Installation

To install the driver provided by the builder in the first place, then the or the cards in the PC.

9.3.1.2 Parameters

The parameters to define are the following :

- Acquisition period

This parameter corresponds to the time separating two position acquisitions. The possible values must be chosen in the proposed list.

- Number of values in tables

Maximum number of values in the sample tables attached to every channel.

9.3.1.3 Usage

In the development workshop

The inputs/outputs must be declared either in a part "global Variables." The analog inputs are of LREAL type.

Example

```
Channel1 ANI0 at: LREAL; // first analog input
Channel2 ANI1 at: LREAL; // second analog input
```

To use them in the program, it is possible to use the variable (ex: ANI0) or its mnemonic (ex: Channel1).

The values are returned in unit according to the hardware.

In the runtime Optima PLC

The inputs/outputs are displayable in the menu "PLC - > Variable - > Analog input ANI" and "Analog output ANO."

Usage of the sample tables

The functions of acquisitions in tables are available while using the IK220 library. Data are stocked in these tables to the asked acquisition frequency.

The tables are named of the following manner :

```
IK220_ < Nb of channel >
```

Thus, the channel 10 will be stocked in the IK220_10 table.

Besides, the "time" is stocked in a table :

```
IK220_ Time
```

The stocked values are the indexes of acquisitions (0, 1, 2, etc.) and the coefficient is defined in order to permit a display in seconds.

9.3.1.4 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Redefinition of the IK220 driver (must be used only one time for a maximum of 8 cards)
2	IK220.dll file not found
3	Failure of the IK220 driver initialization

9.4 ICP DAS

9.4.1 Carte ICP DAS PCI-P16R16

Version 1.2



This driver provides access to 16 isolated inputs and 16 relay outputs of the card ICP DAS PCI-P16R16.

The channel numbers are used as follows :

First channel	Last channel	use
0	15	Isolated inputs of the card
16	31	Output relays of the card

9.4.1.1 Parameters

- Number of the card

The system can contain several card PCI-P16R16. This parameter permits to specify which is addressed here. The number of the first card is 1. This is the default value.

9.4.1.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
1	Bad card number (must be between 1 and 16)
2	Undetected card with given number.

9.5 Phoenix Contact

9.5.1 Modbus / TCP module marks FL IT 24 BK-PAC

This driver allows to drive some Phoenix Contact inputs/outputs using a station of the type FL IT 24 BK-PAC.

The communication between Optima PLC and the hardware Phoenix Contact is performed on an Ethernet network according to the Modbus/TCP protocol.

This driver allows a maximal configuration of 256 DI, 256 DO, 32 ANI and 32 ANO.

To allow the PLC to reach the inputs/outputs, the hardware configuration should be recorded beforehand in the station (with the utilitarian "Factory Line IO Configurator" of Phoenix Contact).

The inputs/outputs settings are the following :

	Optima PLC	Phoenix Contact
Digital inputs	I0 to I255	DI0 to DI255
Digital outputs	O256 to O511	DO0 to DO255
Analogical inputs	ANI0 to ANI31	AI0 to AI31
Analogical outputs	ANO32 to ANO63	AO0 to AO31

9.5.1.1 Parameters

To configure the driver, select it in the list of the drivers then click on the button "Parameters."

There are five parameters but only the parameter "IP Address" is obligatory, the other have some default values :

- IP address :

The IP address must be the one of the Phoenix Contact station.
It is composed of 4 bytes separated by points.
This address must belong to the subnet class C of the PC PLC (Optima PLC).

Example

address of the station head: 192.189.207.60
address of the PC with Optima PLC: 192.189.207.193

If no IP address is entered, an error is displayed at the start of the application.

- Minimum scan period (in ms) :

Period to which the read/write of the inputs/outputs must be performed.

- Delay of the watchdog (in ms) :

Delay of watchdog triggering in the Phoenix Contact station.
If this delay has the value 0 then the watchdog is deactivated.

- Delay of the monitoring (in ms) :

Delay of triggering of the monitoring in the Phoenix Contact station.
If this delay has the value 0 then the monitoring is deactivated.

- Errors display Yes (1) / No (0) :

If this parameter is 1, then the trouble reports appear in case of communication error.

9.5.1.2 Usage

In the development workshop

The inputs/outputs must be declared either in the part "global Variables" either in the part "Statement" of a block.
The inputs/outputs are of type bool for the digital and type LReal for the analogical.

Example

```
DITest I4 at: bool;
DOTest O260 at: bool: = false;
AITest ANI30 at: LReal;
AOTest ANO32 at: LReal: = 0.0;
```

To use them in the program, it is possible to use the variable (ex: O260)

Example

```
DOTest: = I4;
```

The status of communication can be examined :

Status	Description
0	no error
2	Error plots, the slave's answer is a trouble report (frame sent by the incorrect PLC).
3	Error timeout plots, the request ended because the delay between two characters received too important).
4	Error timeout answer, the request ended because the

	delay without answer was too important.
7	CRC error, the check of the received frame indicates a transmission error
8	Error timeout connection, the connection network has not been established or has been interrupted

One status of error indicate that an error occurred but the communication doesn't halt.

For example, in case of error of connection due to a non supplied station, it is sufficient to put under tension the station to communicate, the status iron then to 0 as soon as a correct exchange is achieved.

This status can be used in the program grâce to the IODriverStatus function (IODrivers library).

Example

```

var
  DOTest O260 at : bool: = false;
  StopCycle   : bool: = false;
  TestStatus  : dint: = 0;
  DriverID    : int: = 0;
end_var;

.....

TestStatus:= IODriverStatus (DriverID);
if not StopCycle
  then DOTest:= true;
end_if;

if TestStatus > 0
  then StopCycle:= true;
       DOTest := false;
end_if;

```

In the runtime Optima PLC

The inputs/outputs are displayable in the slim "PLC - Variable."

9.6 Port I/O PC

9.6.1 Port I/O PC - 8 digital inputs

Version 1.0

This interface permits the usage of 8 digital inputs accessible by an I/O port of the PC. It allows the access to the inputs of numerous cards whose interfaces are not available here.

No initialisation of material is performed here, the interface confines itself to the reading of the port. The initializations possibly necessary can be made before the start of the application.

9.6.1.1 Parameters

The parameters to specify :

- IO base address :

The address of the port containing the 8 input bits (consult the documentation of the card to know this address).

Enter the address preceded of 16# to define an hexadecimal address.

- Inversion mask IO 0 to 7 :

This parameter specifies if the statement of the port must be inverse after reading (0: no inversion, 1: the 8 inputs are inverse).

Enter the value of the mask preceded of 16# to define an hexadecimal value.

9.6.1.2 Usage

In the development workshop

The inputs must be declared either in the part "global Variables" either in the part "Statement" of a block. The inputs are of type bool.

Example

```
DITest I0 at: bool;
SATest SA10 at: bool;
```

To use them in the program, it is possible to use the variable (ex: I0)

Example

```
SATest: = I0;
```

In the runtime Optima PLC

The inputs are displayable in the menu "PLC - Variable."
A diagnosis is available in the menu "PLC - Drivers - Infos."

9.6.2 Port I/O PC - 8 digital outputs

Version 1.0

This interface permits the use of 8 digital outputs accessible by an I/O port of the PC. It allows the access to the outputs of numerous cards whose interfaces are not available here.

No initialization of hardware is done here, the interface confines itself to the writing of the port. The initializations possibly necessary can be made before the starting of the application.

9.6.2.1 Parameters

The parameters to specify are :

- IO base address :

The address of the port containing the 8 output bits (consult the documentation of the card to know this address). Enter the address preceded of 16# to define an hexadecimal address.

- Inversion mask IO 0 to 7 :

This parameter specifies if the statement of the port must be inverse before writing (0: no inversion, 1: the 8 outputs are inverse).
Enter the value of the mask preceded of 16# to define an hexadecimal value.

- Entry on the port after every alteration of output value Yes (1) / No (0) :

This parameter permits to write the value after every alteration or at the time of every scan.

9.6.2.2 Usage

In the development workshop

The outputs must be declared either in the part "global Variables" either in the part "Statement" of a block. The outputs are of type bool.

Example

```
DOTest at O2 : bool;
SATest at SA10: bool;
```

To use them in the program, it is possible to use the variable (ex: O2)

Example

02: = SATest;

In the runtime Optima PLC

The outputs are displayable in the menu "PLC - Variable."
A diagnosis is available in the menu "PLC - Drivers - Infos."

9.7 Simulation

9.7.1 Standard simulation driver

Version 1.0

This driver permits to simulate some inputs/outputs in an application.

2048 digital I/Os can be simulated, every I/O being accessible either as input, either as output. 256 analogical channels can also be simulated.

The principle is the following : rather than the statements of the I/O are gotten from a hardware, they are read here or write in tables that simulate this hardware. The action to modify these tables, for the inputs, has for consequence to modify the core image of these inputs for Optima PLC. For the outputs, the tables permit to get the normally controlled electric statement by the application.

9.7.1.1 Parameter

The driver only comprises one parameter: the suffix of name of access to the variable of simulation (see farther). This parameter is only useful in the case where several drivers of simulation are used, in order to differentiate them. In the opposite case it can be left empty (case by default).

9.7.1.2 Usage in the supervision

The access by the "supervision" overlays the standard Optima PLC supervision, as well as the access via OPC server.

Simulation data can be used in the supervision of several manners.

Absolute access

The access is made via the variable "Sim" followed of the waited variable type.

If a suffix has been defined in the parameter of the driver, this one is added at the end of " Sim ". For example, if the suffix is " P1 ", the variable to use is " SimP1 ".

Digital inputs	Sim.I0 to Sim.I2047
Digital outputs	Sim.O0 to Sim.O2047

The values of the indexes can be variable :

Sim.I [R1]

In this case, the simulation input variable whose index is contained in R1 is used.

Attention

So that the number of channel I , O or AN corresponds at the same number channel in the global numbering of Optima PLC, it is necessary that the driver is placed in first in the list of the drivers. Indeed, Sim.I10 corresponds at the 10th channel in the driver. If the first I/O managed by the driver is the number 64, for example, Sim.I10 corresponds in fact in to channel 74 in the global numbering of Optima PLC.

Access by mnemonic

If some I/O are declared in the program, they can also be used via their mnemonic.

Example

```
I_StartCycleButton   at I10 : bool; // button departure cycle
O_CycleInProgressLight at O23 : bool; // light indicator indicating the cycle in progress
```

These two I/O can be simulated while using the following inputs :

Sim_I_StartCycleButton
Sim_O_CycleInProgressLight

9.7.1.3 Access by shared storage zone

A third application can use simulation data by reaching the "memory mapped file" named "OptimaPLCSim (plus the suffix possibly specified in the parameter).

The structure of the memory zone so accessible is the following :

Digital inputs	2048 bytes	Every byte is 0 or 1
Digital outputs	2048 bytes	Every byte is 0 or 1
Analogical channels	256 actual values	Every channel is coded on 80 bits (format standard IEEE 754)

The total size of the shared zone is therefore of 6656 bytes.

Note: the access to this zone is normally made with the Win32 API, by means of the CreateFileMapping functions, OpenFileMappings and MapViewOfFile.

9.7.2 Driver of simulation Solid Dynamics Motion

Version 2.0

This driver allows to simulate inputs/outputs in an Optima PLC application. it is dedicated more especially to the communication with the software Motion of the firm Solid Dynamics.

2048 digital I/O can be simulated, every channel being accessible either as input, either as output. 256 analogical channels can also be simulated.

The principle is the following : rather than the statements of the I/O are gotten from a hardware, they are read here or write in tables that simulate this hardware. The action to modify these tables, for the inputs, has for consequence to modify the core image of these inputs for Optima PLC. For the outputs, the tables permit to get the normally controlled electric statement by the application.

9.7.2.1 Parameters

The driver includes two parameters : these are in particular the names of the two exchange memory mapped file, used for the communication with the other application Motion.

The first parameter corresponds to the file of input of Optima PLC, the second to the file of output.

9.7.2.2 Usage in the supervision

The access by the "supervision" overlays the standard supervision of Optima PLC, as well as the access via OPC server.

The simulation data access is made via the variables "SimMotionOptimaPLCInputs" and "SimMotionOptimaPLCOutputs", consistent of the waited variable type.

If a mapped file name different than default name was used, the variable name is made up of the prefix "Sim" followed by the file name.

Digital inputs	SimMotionOptimaPLCInput.D0 to SimMotionOptimaPLC.D2047
Digital outputs	SimMotionOptimaPLCOutput.D0 to SimMotionOptimaPLCOutput.D2047
Analog inputs	SimMotionOptimaPLCInput.A0 to SimMotionOptimaPLCInput.A255
Analog outputs	SimMotionOptimaPLCOutput.A0 to SimMotionOptimaPLCOutput.A255

The values of the indexes can be variable :

SimMotionOptimaPLCInput.D [R1]

In this case, the variable of entry simulation whose index is contained in R1 is used.

Attention

For the channel A or D to match the same channel in the overall Optima PLC numbering, it is necessary that the driver is placed first in the list of drivers. Indeed, the channel SimMotionOptimaPLCInput.D10 corresponds to the 10th lane in the driver. If the first I/O managed by the driver is the number 64, for example, D10 is in the channel numbering 74 in total Optima PLC.

9.7.2.3 Access by shared storage zone

An application as Motion can use simulation data by reaching the "memory mapped files" named "MotionOptimaPLCInput" and "MotionOptimaPLCOutput" by default.

The structure of the memory zones so accessible is the following :

Size of the segment	4 bytes	Size of the segment on 32 bits
Digital channels	2048 doubles	Every value is worth 0 or 1
Analogical channels	256 doubles	Actual values

Every "double value" is coded on 64 bits to the format of real standard IEEE 754.

The total size of every shared zone is therefore normally of 18436 bytes.

Note: the access to these zones is normally made with the Win32 API, by means of the CreateFileMapping functions, OpenFileMappings and MapViewOfFile.

Note 2 : in this version, no synchronization mechanism is foreseen, the consistency of data is not assured therefore.

9.8 Wago

9.8.1 Modbus/TCP module 750-341, 342, 352, 841, 842

Version 1.1

This driver permits to pilot Wago inputs/outputs by using a station of the type 750-XXX.

The communication between Optima PLC and the WAGO station is performed on an Ethernet network according to the protocol Modbus UDP (idem Modbus TCP, but based on UDP).

This driver allows a maximal configuration of 256 DI, 256 DO

So that the PLC can reach the inputs/outputs, the network configuration (IP address) should be entered beforehand in the station (with the utilitarian "BootP server" of WAGO).

In case of malfunction, the "Diagnosis" information, available in the runtime, allows to assess the situation. If necessary, this information can be copied (via the clipboard) in an email sent to Optimalog to aid diagnosis.

9.8.1.1 Parameters

To configure the driver, select it in the list of the drivers, then click on the button "Parameters."

There are five parameters but only the parameter "Addresses IP is obligatory, the other have some default values :

- IP address

The IP address must be the one of the WAGO station.

It is composed of 4 bytes separated by points.

This address must belong to the subnet class C of the PC PLC (Optima PLC).

example

address of the station head: 192.189.207.60
 address of the PC with Optima PLC: 192.189.207.193

if no IP address is entered, an error is displayed at the start of the application.

- Communication mode

Indicate if the communication is made in UDP (mode with connection and stream management) or in TCP (telegrams, mode without connection). The UDP mode is more flexible and permits a better management of the timeouts.

- Timeout of re-emission

Time in microseconds after which an order is sent back if no answer is received.

- Timeout of reset

Time in microseconds after which the connection is reset if no answer is received.

- Minimum scanning period (in ms)

Period to which the read/write of the inputs/outputs must taken place. For a sweep to the maximal speed, to put 0.

- Watchdog delay (in ms)

Delay of triggering of the watchdog in the Wago station. The entered value is multiplied by 100 ms. If this delay has the value 0 then the watchdog is deactivated.

- IP address locked (yes: 1/non: 0) :

Indicate if the specified IP address must be locked in the WAGO station.

- Display of the errors (yes: 1/non: 0)

If this parameter is worth then 1 of the trouble reports appear in case of error of communication.

- Updating the outputs:

Two modes are possible.

"Automatic" is the default mode. The outputs are updated at the start of the application, with an initial state to 0.

"On software command": In this mode, when you start the application, inputs only are updated. The outputs are not sent to the module Wago. Starting and stopping the update is ordered by calling library functions IODrivers:

IODriverWagoStartUpdateOutputs start updating the outputs.
 IODriverWagoStopUpdateOutputs stops updating the outputs.

The latter mode allows to stop and then to restart the application without affecting the output states of Wago module, which may be necessary in some applications, such as the Building Management. Attention to this watchdog must be disabled (see parameters above).

9.8.1.2 Usage

In the development workshop

The inputs/outputs must be declared either in the part "global Variables" either in the part "Statement" of a block. The inputs/outputs are of type bool for the digital and type Lreal for the analogical.

Example

```
DITest at I4 : bool;
DOTest at O260 : bool: = false;
AITest at ANI30 : lreal;
AOTest at ANO32 : lreal: = 0.0;
```

To use them in the program, it is possible to use the variable (ex: O260).

Example

```
DOTest: = I4;
```

In the runtime Optima PLC

The inputs/outputs are displayable in the menu "PLC - Variable."

9.8.1.3 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
2	Error of frame modbus
3	Timeout during a frame modbus
4	reply timeout on receipt to a modbus request
7	CRC error in a modbus reply
8	No connection with the module
51	Internal error : initialization of modbus impossible
52	Address IP incorrect or unspecified
60	Error of configuration : inputs or outputs variables are used that doesn't exist in the Wago station.
100	Transmission of the configuration to the Wago module

9.9 Network standard Ethernet

9.9.1 Generic Modbus/TCP module

This driver permits to use in a standard manner the inputs/outputs of an equipment answering the Modbus/TCP or Modbus/UDP protocol. The PC acts as the master.

The distribution of the inputs and outputs is made of the following manner :

Digital inputs	indexes 0 to 1023
Digital outputs	indexes 1024 to 2047
Analogical inputs	indexes 0 to 63
Analogical outputs	indexes 64 to 127

9.9.1.1 Parameters

- IP address

IP address permitting to achieve the distant equipment.

- Communication port number

Number of TCP or UDP port used for communication (normally 502).

- TCP mode or UDP

On Ethernet, the Modbus protocol can be based on TCP (linkage with establishment of a connection) or on UDP (exchange of datagrams without notion of connection).

When it is possible, prefer the UDP mode, that is lighter and permits to Optima PLC a bigger mastery of timing. Attention, this UDP mode is not allowed with a Hilscher board.

- Hilscher board number

If the communication is managed by one of the 4 Hilscher cards present CIF in the PC (it must be about a CIF card 50-IN), to specify his/her/its number here, otherwise to leave 0.

The corresponding driver to the card Hilscher CIF 50-SOME in mode master must be configured previously in the list of the drivers.

Attention, this parameter is not processed if the UDP mode is selected previously.

- Scanning period

This parameter is not managed currently. Leave the value 0.

- Display of the errors

If the value Yes or 1 is specified, a defect of communication will be signalled to the display. Otherwise, such a defect is only indicated in the log file.

- Modbus function for reading the digital inputs

This setting adjusts the function used to the possibilities of the equipment. FC1 to FC4 functions are allowed. Using FC3 or FC4 functions is optimal.

- Number of digital inputs available on the equipment

Indicate here the number of digital inputs available on the equipment. this number (128 maximum) indicates how many inputs will be read at every cycle by the driver.

- Address of the first digital input

Address of the first input to read (in the space of the words 16 bits).

- Modbus function for writing the digital outputs

This setting adjusts the function used to the possibilities of the equipment. FC15 and FC16 functions are allowed. Using FC16 function is optimal.

- Number of digital outputs available on the equipment

Indicate here the number of digital outputs available on the equipment. this number (128 maximum) indicates how many outputs will be written at every cycle by the driver.

- Address of the first digital output

Address of the first digital output to write (in the space of the words 16 bits).

- Modbus function for reading the analogical inputs

This setting adjusts the function used to the possibilities of the equipment. FC3 and FC4 functions are allowed.

- Number of available analogical inputs on the equipment

Indicate here the number of available analogical inputs (or counters) on the equipment. This number (64 maximum) indicates how many inputs will be read at every cycle by the driver.

- Address of the first analogical input

Address of the first analog input to read (in the space of the words 16 bits).

- Modbus function for writing the analogical outputs

Only the function FC16 is allowed here.

- Number of available analogical outputs on the equipment

Indicate here the number of available analogical outputs (or counters) on the equipment. This number (64 maximum) indicates how many outputs will be written at every cycle by the driver.

- Address of the first analogical output

Address of the first analogical output to write (in the space of the words 16 bits).

Precisions concerning the numbers of inputs and outputs.

The driver creates the frames of reading and writing according to the parameters mentioning the numbers and addresses of the inputs and outputs. There are 4 defined frames (digital inputs, digital outputs, analogical inputs, analogical outputs) therefore maximally.

The total cycle time being function of the number of frames, it is preferable to leave to 0 a number of variables if these are not used in the application. For example, if the application doesn't use any analogical outputs although there is some available, leaving to 0 the number of analogical outputs will avoid the creation of an useless frame and will reduce the scanning time of the useful variables.

9.9.1.2 Status values

Status	Description
-1	Uninitialized driver
-2	Driver halted in phase of stop of the application
0	Nominal performance
2	Error of frame modbus
3	Timeout during a frame modbus
4	reply timeout on receipt to a modbus request
7	CRC error in a modbus reply
8	No connection with the module
51	Internal error : initialization of modbus impossible
52	Address IP incorrect or unspecified
60	Error of configuration : inputs or outputs variables are used that doesn't exist according to the parameters entered.

10 Versions history of the software components

The main modifications of every version are listed here.

10.1 OptimaPLCRuntime.exe (runtime Optima PLC)

10.1.1 V1.4.10

- Management of the virtual keyboard, with automatic apparition. Option in the menu Controls. Option in the menu window in the case where the automatic option is not nicked.
- Automatic call of the main page of supervisory associated to the application.
- Creation of a diagnosis Debug Data.xml file
- Taken in account of the jitter for the interval of execution of the tasks

10.1.2 V1.5

- Addition of one specific page for the visualization of the DT temporizations.
- Optimization of the cycle time for the scrutation modbus and modbus / tcp.

10.1.3 V2.0

- Taken in charge of the loading of the online modifications.

10.1.4 V2.1

- Taken in account improved of the modifications of mnemonics at the time of an online modification.

10.1.5 V2.2

- Working in mode reduced without the kernel real time.
- Possibility to adjust the period of the tasks of priority 8 of 2 ms to 20 ms (impact also on the management of the exchanges of E /S managed by Windows functions to this level of priority).

- Visualization ladder improved (dynamic display of the valid paths at the time of the execution).
- ModBus / UDP protocol added to the protocols serial ModBus and ModBus / TCP.

10.1.6 V2.2.1

- Correction of a problem occurring at the time of the closing of the application on some PC

10.1.7 V2.5

- Various optimizations for working without the real-time kernel.

10.1.8 V2.6

- Possibility to visualize the functions in the débbuger (the visualization was limited before to the function blocks and to the programs).

10.1.9 V2.8

- Correction of a display problem sometimes too long in the explorer of variables.

10.1.10 V2.8.2

- Correction of a display problem of display of the tasks data

10.1.11 V2.9.4

- Visualization of the active step of an Apigraf sequence in the "Hint" displayed at the passage of the mouse.

10.1.12 V2.9.5.5

- Correction of a bug making the PC reboot at application start in some special situations.

10.1.13 V2.10

- Dynamic graphic view of Apigraf grafcet

10.1.14 V2.10.8

- Adding the variable UTC to obtain the time in Coordinated Universal Time (time independent of the seasonal time change).

10.1.15 V2.10.9

- Changes in management retain files on error format (adding a CRC calculation).

10.1.16 V2.11

- Change in ini files management (no systematic write at end of execution or if there is no change).
- Debugger change (minimum size tree can be hidden).
- Ability to view data exchanged on the serial ports.
- Lists of expression: from 4 to 6.
- Ability to initiate a search in the workshop from the editor ST or Apigraf.

10.1.17 V2.12.7

- Adding the options "manual commands" in the menu, available when using the library "manual".

10.1.18 v2.13

- New method of dynamic tracking of active steps in SFC.
- Followed variables (mode activated by a compiler option).

10.1.19 v2.14

- SFC Views : dynamic tracking of active steps in SFC disabled by default.

10.2 Drivers.exe (selection of the drivers in Optima PLC)

10.2.1 v1.4.9

- Displacement of the documentation of the drivers in DriversOptimaPLC.chm, addition of the management of the .chm files.

10.2.2 V2.3

- Possibility to put a standard driver in mode "simulation."

10.2.3 V2.10.7

- Hilscher borads : Adding functions to import configuration from SyCon.

10.2.4 V2.10.8

- Added drivers for cards Advantech PCI-1733 and PCI-1734.
- Fixed critical for access to analogue channels via the driver station connected to a generic Hilscher card.

10.2.5 V2.12.7

- Corrections in importing data from Hilscher SyCon.

10.2.6 v2.13

- Changes in the driver Modbus/TCP Generic: added selection for Modbus functions used.

10.2.7 v2.14

- Added drivers of Advantech Adam modules: 4017, 4018, 4050, 4053, 4055.
- Correction in the management of Modbus/TCP.

10.3 OptimaPLCWorkshop.exe (atelier Optima PLC)

10.3.1 V1.4.9

- Integration of the supervisory in the workshop Optima PLC.

10.3.2 V1.4.10

- Integration of the Apigraf language in the POU's.
- Addition of the tab "structures" with visualization of the overlapping of the POU's bound to the declarations
- Improvement of the selection of the objects in the diagrams ladder.
- Addition of contextual menus of edition in the publishers text
- Correction minor various

10.3.3 V1.5

- Addition of the ModBus library
- Operation to copy / glue in the publishers ladder and FBD.

10.3.4 V1.5.1

- Modification of the management of the DT temporizations: passage in Time, addition of the MilliSecTime conversion function.
- Edition of the SFC diagrams: a problem empêchait in some cases the insertion of a return if all vertical branches of a group ended by a return.
- Possibility to express characteristic constant values: INT#5, WORD#32, etc...

10.3.5 V2.0

- Modification of the on line application.
- Various improvements in the management of the SFC actions of a SFC diagram.
- Addition of new function blocks and functions.
- Corrections of various minor bugs

10.3.6 V2.1

- Various improvements in the generation of the code ladder.

10.3.7 V2.1.2

- Addition of the library Emails, permitting the sending of emails by programming.

10.3.8 V2.1.3

- Addition of a function of license modification by exchange of code, accessible since the window of consultation of the licenses.

10.3.9 V2.2

- Possibility of deactivation of the kernel real time in the general options of compilation.
- General improvement of the generation of the code ladder.

10.3.10 V2.2.1

- Correction of a scrolling problem with the wheel of the mouse under edition in the diagrams ladder and function blocks.
- Modification of the management of the licenses (taken in account of the different versions).
- Addition of the compatibility ChnXXX functions with Apigraf (conversions of numeric values in characters strings).
- Addition of the functions of Time_To_String conversions, Date_To_String, Time_Of_Day_To_String and

Date_And_Time_To_String.

10.3.11 V2.3

(15/2/2007)

- Addition of the management of the users
- Addition of the possibility to execute and to modify on line the application on a distant PC.
- Syntax of programming: .bitX suffix or .bit [X] usable with all whole types to test or to modify an internal bit to a variable, an expression or a constant.

10.3.12 V2.4

- The libraries memorize the libraries of which they depend.
- Verification of the identity of size of the variables declared in absolute with the AT (the declared variable and the variable préexistante overloaded must have the same size) syntax.
- Correction of a problem of displacement of object in the publisher ladder (the displacement of an initially empty zone provoked an error).
- Correction of a problem of edition of the general configuration (an error was signalled if Apigraf was not installed).

10.3.13 V2.5

- Correction of a problem empêchant the correct working in the case of a mnemonic on alarm or temporization in Apigraf language.
- Real-time kernel disactivated by default.

10.3.14 V2.6

- Possibility to define the variables of ANY_INT type and ANY_REAL.
- New PCI1784U library.
- New SharedMemory library.

10.3.15 V2.7

- Addition of the TOWriteCSV functions, TOWriteLnCSV, TOWriteTab, TOWriteLnTab in the TEXTOUT library.
- Improvement of the system of prévisualisation and impression of the programming file.
- Corrections of various minor problems.

10.3.16 V2.8

- Corrections of various minor bugs.
- Correction of a memory flight in the edition of the POU's.
- Addition of new libraries (PCI1243U, THERMACAM, PROSILICAGIGE) and new functions in the existing blocks.
- New improvements of the impression of the programming file.
- Automatic visualization of the interface of the blocks overflowed by the mouse.
- Possibility to program the actions in situ of the SFC diagrams in Apigraf or in IL.

10.3.17 V2.8.2

- Addition of drivers (boards Advantech PCI-1747, PCI-1723, PCI-1780).
- Addition of a library of management of the TC315PCI (3C Concept) board
- Modifications of the driver of the Hilscher generic CIF board
- Correction in the syntax of the sequences in Apigraf language (problem on the word key SEQUENCE).

10.3.18 V2.9

- The use of function block in diagram ladder or diagram function block, don't require a systematic declaration of the blocks anymore in the part declarations. For it, the name of variable of the block must be left empty. A name of variable is allocated by default then to the block.
- Addition of the driver and the library for the board IK220 Heidenhain.
- Additions of various functions in the libraries standards.
- Addition of the driver for board Hilscher CIF50-COM dedicated to the CAN communication layer 2 (free protocol).
- Multiple minor bug corrections
- Modifications and improvements of the management of the mode multiprocessor. Optimization of the times of treatment and commutation.

10.3.19 V2.9.1

- Addition of the FILTERS, SLOPES, LINCURVE libraries.
- Correction of a problem affecting the declarations of Apigraf sequences (tie possibly not created with the

sequence if this one is the only used variable).

- Modifications for compatibility with Windows Vista
- Modification of the format of the libraries to integrate some bilingual information of use.

10.3.20 V2.9.4

- New option in the properties of the POUs permitting to use a function without parameter nor return type as a macro from the supervisory.
- Possibility to test the activity of a SFC sequence step from the outside of the block (syntax: InstanceBlocName.StepName.X).
- Correction of display problems in the textual editors.
- Integration of the Digimétrie driver USB-ROPTO16.
- Integration of new functions of for interaction with Windows and control of the runtime display (SYSTEM library).
- Modification of the behavior in case of compilation error (possibility to continue or to stop the compilation).
- Corrections and various minor improvements

10.3.21 V2.9.5

- Corrections in functions ValReal, ValLReal, ValInt, ValDInt, ValLInt
- Correction in function FileReadLn
- Test for availability in download of a new version from Optimalog website

10.3.22 V2.9.5.5

- Minor corrections
- Library FTPClient added, some new functions in other existing libraries
- Corrections in char type management (assignments)
- Driver for PCI-1762 (Advantech) added

10.3.23 V2.10

- Corrections and various improvements concerning the Apigraf language.
- Possibility to visualize the resulting diagram of an Apigraf sequence
- Periodic automatic save of the project in progress in subdirectories of the "Backup" directory situated in the project.
- Addition of the support of Optima Datalog (Datalog library) allowing a database access by SQL requests.
- Various additions in the standard libraries.

10.3.24 V2.10.1

- Correction of a bug preventing modifications canceling in SFC and in ladder.
- Improvements concerning the visualization of the values of the expressions in Apigraf

10.3.25 V2.10.6

- Automatic transmission system of internal diagnostic data (improving software quality).

10.3.26 V2.10.7

- Correction in the internal management of the date and time (sync with the Windows clock).

10.3.27 V2.10.9

- Critical correction in internal timings management to avoid a deadlock.
- Changes in the SFC editor to see the steps to cause a return.
- Management of function blocks declared in an array or structure.

10.3.28 V2.11

- New menu option "re-open" a project.
- New library: 7Zip (compression and decompression of files)
- Adding search capabilities file (Files library).
- Possibility to run the workshop with a runtime license in "read only" mode.

10.3.29 V2.12

- Possibility to choose the default project directory

10.3.30 V2.12.7

- Add the library "Manual" and the ability to manage manual commands.
- Fixed a problem in "online change" management, in the case of real-time kernel disabled.

10.3.31 v2.13

- Optimization of the process of runtime code generation, increasing the speed of compilation.
- Various fixes in the compilation and remote execution.
- Compiler option to add the tracking code variables (for development and debug).

10.3.32 v2.14

- Corrections in the display of SFC charts
- Various minor fixes and improvements

10.4 SuperGW.exe Optima View (Supervisory workshop)

10.4.1 V4.53

- Addition parameter "automatic Closing" at the end of n seconds in the supervisory forms

10.4.2 V4.54

- Correction problem annulment definition of the objects of supervisory.
- Addition object of supervisory "Display Hour"
- Addition object of supervisory "Display dates"
- Addition object of supervisory "Calendar"

10.4.3 V5.0

- Access to the variables OPTIMA PLC or APIGRAF directly with the TCP / IP sockets
- Possibilities to connect to the OPC server of the market
- The complex expressions are decomposed directly in the supervisory
- Addition menu "Diagnoses" in the main menu of the supervisory workshop.

10.4.4 V5.1

- Addition object "Arrows Loud / Low"

10.4.5 V5.2

(09/02/2007)

- Addition menu "Tools - > Administrator of alarms." This menu throws or remind to the screen the administrator of alarms centralized: OPTIM'ALARM
- Correction deletion table in the object "Curves"

10.4.6 V5.3

(01/03/2007)

- Corrections object curves with samples tables

10.4.7 V5.4

(07/03/2007)

- Internal modifications of all Listviewses in StrindGrid for the XP look

10.4.8 V5.8

- Modification of the objects curves
- Addition of the object "button in LED"
- Various minor corrections

10.4.9 V5.10

(07/01/2008)

- Access direct memory of the local samples tables
- Possibility to define some doorsteps in the curves - min and max paramétrables with an expression on the curves
- Gradients (damaged) on the curves

10.4.10 V5.11

(07/01/2008)

- Possibility to define some doorsteps in the curves - min and max paramétrables with an expression on the

curves of the tape recorders

- Gradients (damaged) on the curves of the tape recorders
- Explorer of OPC server
- Addition of the object "Sizes up circular"

10.4.11 V5.12

(26/03/2008)

- Addition of the "Cursors / TrackBar" tabs with new TrackBarses paramètrables
- Window of seizure with unwinding list: Addition of the option "Affectation of the index of the element selected to the expression." Correction various on the object.
- Addition of a function button in the tab button: Button tidy appearance with integration of an icon that permits to call one macro

10.4.12 V5.13

(31/03/2008)

- Update helps on line

10.4.13 V5.14

(08/04/2008)

- Object window of edition: Possibility to display the types time, date, time_and_date, time_of_day of optima PLC to the format Dates / hour and to seize some date / hour values in the window.
- Object window of special edition: Possibility to display the types time, date, time_and_date, time_of_day of optima PLC to the format Dates / hour and to seize some date / hour values in the window.
- Addition of the object Window of edition Dates / Hour that permits to seize of dates / hours with automatic opening of a selection calendar for the Types date and of seizure specific Hour / Minutes / Seconds for the Time types
- Object Folder: memorization of the order of the pages with the safeguard with the possibility to specify the index of page of the folder opened by default.
- Object Calendar: possibility to supervise expressions of type dates or date_and_time coming from optima PLC as for the DateTimePicker object
- Addition of the Menu Diagnoses - > Samples tables bound in memory that permits to visualize a window of diagnostic correspondent. This window permits to see the general data coming from the tables of measure bound in memory.

10.4.14 V5.15

(17/04/2008)

- Virtual keyboard: possibility to have an automatic virtual keyboard on the window of seizure. Addition of the option "automatic" virtual Keyboard in the menu: Options - > general Options
- Objects windows of seizure, special seizure, window to unwinding list: Addition of parameters bound to the virtual keyboard. Possibilité to disactivate the virtual keyboard in particular for a window of seizure and to specify the type of virtual keyboard (alphanumeric or numeric)
- Addition of a palette of functions of alignment of the objects in the workshop
- Modifications icons of the main menu

10.4.15 V5.16

- Modification of selection of projects form
- Directory of projects is now "Projects" and no more "Projets"
- Directory of examples is now "Examples" and no more "Exemples"
- Addition of object circular gauge with needle
- The standard cursor object functions with Real variables
- The Scroll bar object functions with Real variables
- Object gauge, meter : the values min and max function with real variables
- Management of the protective USB Optimalog dongle for autonomous version
- HINT : possibility of input on several lines in a memo
- Addition translucent function button object
- List of the local macros: Ordering of the macros in the alphabetic order
- Addition of "EXCEL link" object and new tab of EXCEL objects

10.4.16 V5.17

- Modification of main menu (Size of the buttons)
- Improvement of diagnosis window of the OPC client - details on the server
- Addition of local macro DIAGCLIENTSOPC that permits to diagnose the links with the OPC server

10.4.17 V5.34

- Added new object type indicator 1
Adding new push button type 1
- Fixed window resizing settings shape object
- Helpfiles bilingual
- Adding object event handler in "Other"
Used to trigger macros on rising or falling value of an expression or boolean variable.
- Fixed cut objects in a panel
- Fix multiple selection of objects
- Object Button PNG adding square types yellow green blue and red
- Object seeing PNG adding square types yellow green blue and red
- Object Area click: Adding expression activated monostable or bistable
- Object Recipe Correction: If a directory is specified for loading a recipe.
- Object Glow Button: Fix resource consumption
- Object Glow Button: Added expression by activated monostable button.
- Object Glow Button: Added expression by activated monostable button.
- Object Button Windows: Adding expression by activated monostable button.
- Palette alignments: Adds buttons to move selected objects by 1 pixel up, down, left and right
- Corrections for execution Macros draft area click, various buttons
- Adding local macros:
HideCursor: hides the mouse cursor in the supervision (needed for touch screens)
ShowCursor: redisplay the mouse cursor in the supervision
HideSuperWindow: Cache Page supervisors passed as parameter or the current page
ShowSuperWindow: Displays the supervisors passed in parameter
BringToFrontSuperWindow: Sets the foreground page to supervision
- Fixes bug on advanced trackbar object - problem with very large values of MIN and MAX ranges from very small scale.
- Fixed flickering object Display Date and Object Viewing Time
- Fixed Buttons for Glass and Glow icon justified + refreshment low notes on the button Glow
- Added option "mono / bistable" on purpose "pushbutton"
- Input objects: the tab key no longer takes into account the entry windows that are ReadOnly
- Adding Macro ReloadImages which causes the file to the image object to be reloaded. Scans all pages open supervision.
- Adding the Windows Media Player object
- Image Subject: Adding a property "Maintain image proportions"
- Adding macro: CopyPageBitmap (C: \ CURVES) CopyPageBitmap (\ CURVES) CopyPageBitmap
- Create a bitmap copy of the current page of supervision in the current directory if no parameters or in the directory passed as a parameter eg C: \ CURVES rep or under the project eg \ CURVES
- Fixed virtual keyboard: problem when changing page
- Fixed CPU time consumption with strings in the edit window standard
- Fixed CPU time consumption with strings in the edit window special
- Object Browser Enhancements
- Ability to name objects (without spaces and special characters)
- Possibility to force an invisible object and all sub objects (useful for panels, ..)
- Possibility to cancel forcing invisible to an object and all its sub objects (useful for panels, ..)

- Update Macro COPYPAGEBITMAP
- Updated online help Media Player object
- Adding input X and Y size of the window of supervision
- Added option "Stayontop" window always in foreground in the options page of supervision.
- Integration of virtual keyboard to input a password opening page
- Added macro EnableVirtualKeyboard
- Added macro DisableVirtualKeyboard
- Image Correction Re: "Keep my image proportions" with option "Adjust image size to the size of the object. Also works with the "centered image"
- Correction option window can be closed "pages supervision. The window is closed with RuntimeClose or Close

10.4.18 V5.38

- Fixed keyboard : Position adjusted
- Possibility to define an opening position for the supervisory pages
- ListBox Object: A change in the value of the expression associated with the object modifies the selected item.
- Adding object Flash Player to play files Adobe ShockWave Flash

10.5 ServAPIOPC.exe (OPC server Apigraf IP)

10.5.1 V5.31

- Integration of the Comp type

10.5.2 V5.32

- Internal modifications (Infos of the file)

10.5.3 V5.50

- Remove the requirement to precede the items of the local IP address.
- Ability to predefine items with a type different from the default string type.

10.6 OptimaAlarm.exe (Optim'Alarm)

10.6.1 V1.1

- TypeAlarme addition: Writing YES / NO in a file
- TypeAlarme addition: Temporization of hold in account alarm

10.6.2 V1.2

- Possibility to associate an icon to a type of alarm. The icon appears then in beginning of the alarm line
- Files automatic monthly texts

10.6.3 V1.3

- Management sending of email: To configure emails first in the menu Configuration - > Sending of emails then to specify by default in the definitions of the alarms the name of the dest
- Correction button user to use the parameters (example: call of a supervisory form)
- Non protected trial version: function solely with OPTIMA PLC and no with the OPC server of the market

10.6.4 V1.5

(29/01/2007)

- Management of the users - definition of the users among several groups. Possibility to give some rights for the discharge or the modification of the definitions of alarms
- If one executes a second time the software, the executable thrown déjà gets forward plane
- Relief on the seers of groups of alarms
- Management of the protective key

10.6.5 V1.6

(14/02/2007)

- Automatic starting parameter
- Writing Login / Logout infos in file text
- At the time of the creation of a new alarm, the initial fields are duplicated since the selected (if an alarm is selected in the list) alarm

10.6.6 V1.8

(07/03/2007)

- Addition option starting of the main window minimized if automatic starting of the application
- Sorting of the alarms order of priority then date apparition
- Update window diagnoses sockets data PLC.

10.6.7 V1.9

(13/04/2007)

- Possibility to have a headband always visible at the bottom of the screen. This headband displays the last active alarm.

10.6.8 V1.11

(14/04/2008)

- Optim'Alarm can function in Mr. or in Slave. If the mode slave is activated, Optim'Alarm slave connects on an Optim'Alarm designated by his/her/its IP address, these are then the master's alarms that are displayed and the user fear to acquit according to the definite rights the master's alarms.
- The mode slave doesn't require a license.
- Modifications aspect visual main window.
- Corrections to main window refresh with a click on the OptimAlarm icon in the "tray"

10.6.9 V1.12

- Alarms groups : aspect improved

10.6.10 V1.20

- Various minor amendments.
- Ability to assign an alarm to two different groups.
- Fixed a problem with the icons associated with alarms, occurring when the icon is not available.

10.6.11 V1.23.1

- Possibility to submit the activity of an alarm to an external variable.
- Limiting the number of occurrences of the same alarm.

10.7 Tracks.exe (Optima track)

To splice topic text wretch.

10.7.1 V1.0

(28/12/2006)

- Creation of the software

10.7.2 V1.1

- Addition of the management of the users.
- Expressions for the definition of the outputs.
- Indicatory output variables of the working state.

10.7.3 V1.4

Version including the management of 200 tracks.

10.7.4 V1.5

Possibility to import or to export the configuration of the tracks since or toward a CSV file.

10.7.5 V1.5.1

Correction of an error affecting the safeguard of the configuration.

10.7.6 V1.6

Two new types of track forçage: until specified one date / hour, and permanent.

10.7.7 V1.7

Ability to get the links variables directly from the available OPC or Optima PLC variables.

10.7.8 V1.8.1

Ability to view the schedule for a full year.

10.7.9 V1.9.1

Ability to change the schedule remotely from a client with no license Optima Tracks.

11 Frequent questions

The questions the most frequently asked are listed here with their answer.

11.1 General questions

11.1.1 Why use the PC as a PLC ?

With a PC, you have a user-friendly interface operator permitting to follow the process in progress and very simple to develop.

It also gives you storage capacities required when a tracking is necessary.

The PC is an open architecture and standard. You are not forced to use the proprietary hardware and can put in competition the suppliers of PC, especially as the offer of the market is very large.

Finally, the usage of a same platform for the development and the execution of the application procures a certain comfort for the clarification, the enabling, the maintenance.

11.1.2 Does Optima PLC permit to program a PLC of the market?

No, Optima PLC is foreseen and optimized to function only on a PC basis.

Future developments are planned however to permit the transportable code generation (C and C++), and executing itself on other OS than Windows.

11.1.3 Is Optima PLC a simulation tool ?

Optima PLC can be used for simulation and formation; in particular, the fact that it functions on PC facilitates its integration a lot within a set of other software.

But Optima PLC is before everything a true system of PLC on PC, for an industrial usage.

11.1.4 Questions concerning the real-time aspects

The real-time aspects of Optima PLC are an important point of the product, and generate numerous questions.

11.1.4.1 Is Optima PLC a real-time system?

Yes, the Optima PLC runtime uses for its execution a real-time kernel conceived and developed specifically by OptimaLog, to get under Windows a regularity and an optimized and adapted determinism to the automatic device.

The gotten performances permit to work with a minimal period of 1 ms, and a maximum jitter of the order of 20 μ s.

Contrary to some products competitors that pretend to real-time, Optima PLC is based therefore on a low-level software layer that preempts the Windows kernel, and not only on Windows multi-tasking scheduler.

However, as with all real-time systems executing over Windows, it doesn't make Windows a real-time OS. It is for example not possible to call functions of the Windows API from the real-time tasks of Optima PLC.

11.1.4.2 Can Optima PLC function without the real-time kernel?

Yes, the real-time kernel is an option. Its usage is one of the compiling options.

The real-time kernel is deactivated besides by default.

11.1.4.3 Why is the real-time kernel deactivated by default?

The performance of the real-time kernel requires important resources (memory in particular). These are not always available on some PC :

- Old PC
- PC used to develop, but also for other usages
- PC with few of resource
- PC used to evaluate Optima PLC, but non dedicated to the performance of the application

To avoid all problem, the real-time kernel is deactivated by default from the version 2.5.

11.1.4.4 Does Optima PLC require the supplementary hardware to function in real time?

No, Optima PLC only uses resources of a standard PC.

11.1.4.5 I did think that it was possible to make real-time with Windows, I don't see what Optima PLC brings besides?

Windows is not designed for making "hard" real-time. If performance can be excellent, the very structure of Windows does not guarantee the consistency of them with a classic application.

The fact that Windows is a preemptive multitasking OS, which actually is necessary to make real-time, is not enough. Other conditions are required on the job management and their commutation, the structure of drivers, management of the expanded memory, etc. ...

The confusion is often managed by some market players who try to accredit the idea that their product guarantees real-time performance. It is also because of some elements of Windows itself : for example, the level of the highest priority that can be attributed to a task called "TpTimeCritical" while, as for all the tasks carried out at the user level Windows, there can be no question of genuine real time.

This inability to work in real time under Windows explains the emergence of a parallel market of truly real-time systems, based solely on software solutions or integrating additional equipment, and able to integrate more or less to Windows.

11.1.4.6 Why not use a real-time system of the market rather?

If you wish to make "classic" automatism applications, Optima PLC offers some advantages over real-time systems also proposed :

- Optima PLC is a tool dedicated to automation, it does not require sophisticated engineering competence developing specialized in real-time systems. A short training course or a self-sufficient in its grip.
- Optima PLC is lightweight, simple and easy to install and use.
- Optima PLC is much cheaper to buy and use than a real-time system, albeit powerful, but oversized and too complex for automation applications, although it is possible to conceive with this tool applications usually confined to complex real-time systems.

11.1.4.7 Does Optima PLC permit to make real-time with all PC?

Typically, there is no problem with standards uniprocessor PCs. However, some PC functions installed can disrupt the real-time functions of Optima PLC. On the other hand some multiprocessor PCs does not currently operate properly the kernel real-time, although this problem remains marginal.

The problems usually come from one or several drivers equipment installed under Windows, unconnected with Optima PLC, and "hiding" (disabling) sometimes the system interrupts for a time too important. There is then an increase in the "jitter" (ie variation) timing of the tasks of the PLC.

This problem is common to all real-time systems operating jointly with Windows. It requires you to perform some tests to measure the maximum jitter observed. In most cases it remains below 20 μ s.

On the other hand, the real-time kernel of Optima PLC requires substantial resources in memory, which can sometimes be a problem on some PCs used for other purposes, and which run many other applications.

Optima PLC can show the performance, particularly changes over the period of execution of the tasks of the PLC. In the event of disruption too high, a warning message is displayed.

11.1.4.8 Does Optima PLC operate resources of the PC multiprocessors?

The multiprocessing mode is operated effectively, but differently depending on whether the real-time kernel is activated or no.

If the real-time kernel is activated : all real-time tasks are executed on a same processor. The tasks managed to the Windows level (priorities 8 and 9, displays) are executed indifferently on every processor.

If the real time-kernel is deactivated : all tasks are managed to the Windows level, and can execute indifferently on every processor.

11.1.4.9 Does Optima PLC permit to make real-time with all interfaced hardwares?

No, the performances are not the same with all hardwares.

To create the interface with a hardware, two solutions are possible in the setting of Optima PLC :

Usage of standard Windows drivers

These drivers are relatively simple to interface with Optima PLC, but their use imposes the usage of the Windows API, with which it is not possible to make the true real time: it is impossible for Optima PLC to guarantee a time of reaction because the sequencing is managed directly by Windows.

The drivers currently based on a TCP/IP connection network with Ethernet use the network layer of Windows and don't procure any real-time performances.

The performances achieved by this type of driver are however very optimized in Optima PLC. The typical cycle time (time separating two actualizations of the inputs/outputs) is of about 10 milliseconds, with maximal variations of some milliseconds.

Usage of specific drivers to Optima PLC

The drivers developed by Optimalog generally permit to reach the hardware directly and to use it with the maximum possible performances. It is generally the case of the cards or facilities of which the specifications for direct access to the hardware are documented by the providers, as for example Hilscher.

The performances of these drivers can achieve the maximum allowed by Optima PLC in term of cycle time, that is normally the millisecond. These performances are however also dependent of the hardware.

11.1.4.10 What performances does one get without the real-time kernel?

The base period is generally one millisecond, but can vary up to 3 milliseconds: the regularity is much worse than the real-time kernel. However, the period obtained is quite sufficient to run many applications.

11.1.5 Is it possible to pass from a language of the standard automatically to another?

No, and that possibility has never been considered by designers of the standard unlike a tenacious myth.

Some portability would be possible if a very small part of the programming possibilities was used in each language, but taken many difficulties. It is not currently a priority in Optima PLC.

11.1.6 Is it possible to modify an application without stopping it?

Yes, Optima PLC permits to make important modifications in an application and to put them on line with a lock time lower to the millisecond.

For more of information, to see ["On line modifications"](#)^[376].

11.1.7 I did already work with Apigraf, can I migrate an Apigraf project easily under Optima PLC?

If both tools are based on the same general principles, some differences make that the migration from Apigraf toward Optima PLC is not an immediate operation. This one remains however very simple.

The basis Apigraf language (combinatory) also exists in Optima PLC, however some adjustments must be performed by hand (to see [Differences with the language original Apigraf](#)^[372]).

- The declarations, in particular the definitions of mnemonics, must be redefined in accordance with the standard IEC 61131-3.
- The existing software modules in Apigraf don't have all their equivalent in Optima PLC.
- Some minor differences, in the types of variables for example, must be taken in account for the migration.
- The IHM part must be redone (man / machine interface) in Optima PLC.

11.1.8 Where can I download the last version of Optima PLC?

You can update Optima PLC in the following site :

Web site

<http://www.optimalog.com/downloadopl.html>

Site FTP OptimaLog

<ftp://optimapl@optimalog.serveftp.net>

11.1.9 Does Optima PLC respect the standard IEC 61131-3?

Optima PLC is one of the only software of its category that respects the definition set of the standard and its recommendations.

11.2 Technical questions

11.2.1 Hardware

11.2.1.1 What configuration is necessary to make Optima PLC operate ?

The following configuration is advised to run the runtime Optima PLC.

PC with a Pentium IV, 2 GHz
Windows XP Service Pack 2 or Windows Vista
One USB minimum (key protection)
1Go RAM minimum
Filesystem NTFS advised

The workshop development can operate on a platform standard Windows XP SP2 or Windows Vista.

11.2.1.2 What can arrive if I don't have memory enough on the PC?

To get some real-time performances at the runtime execution, Optima PLC must lock a big physical amount of memory. If this actual storage is insufficient, it can disrupt the real-time performances and in some cases provoke a crash of Windows.

At the time of the start, the runtime Optima PLC indicates in the log file (Events.log) if the quantity of RAM is insufficient to make operate the application.

11.2.2 How to make...

11.2.2.1 How to get some information on the user's license of Optima PLC?

In the workshop or in the runtime Optima PLC, the first menu comprises an option "License" that provokes the display of information on the user's license.

11.2.2.2 How to limit the hazard that the user corrupts data or the system under Windows?

When a PC is put in production with an application running under Windows, the risk is high that some users go modify data or operating parameters of the system so reckless.

To avoid this risk, it suffices that the system works in a session "no administrator", with limited rights. However, the application can run in this type of session, because access to the resources of the PC it is then prohibited.

To overcome this problem, simply set the application to run in the context of a user with administrator rights. To make this parameterization, see [Windows session for execution](#)^[28] and [Parameterization of the Windows sessions](#)^[29].

11.2.2.3 How to prevent the user to change operating parameters in the runtime?

You must activate the mode "[management of the users](#)^[28]" in the runtime.

11.2.2.4 How to use timers in Optima PLC?

The different possibilities are described in the paragraph [Techniques of time management](#)^[37].

11.2.2.5 How to display some warning messages for the user?

See the functions library "Messages", that contains the functions permitting to display the information messages, errors, confirmation demands... See the documentation also of advanced programming.

11.2.2.6 How to force a SFC diagram to its initial step ?

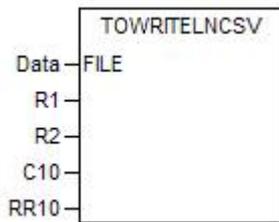
See the block ResetSFC.
Note that this possibility is not considered in the standard.

11.2.2.7 How to write data in a file to reread them with a spreadsheet?

This operation is very simple.

- Declare the file in the menu Options / Output text files.
- Load the TextOut library in the standard libraries (tab "libraries").
- In the program, to use the TOWriteLnCSV functions (to generate a file with data separated by commas) or TOWriteLnTabs (to generate a file with data separated by tabulations).

Below, in the example one writes in the file Dated (the real name on disk is defined in Options / Output text files) the values of the variables numeric integer R1 and R2, of the C10 character string (the value will be framed of characters") and of the real variable RR10. The values separated here by commas (CSV format), a TSV format (tabulations) is also available with TOWriteLnTab.



11.2.2.8 How to enter values of parameters on my touch panel?

To see the option "[automatic virtual keyboard](#)"^[324] of the runtime, that permits to make appear a keyboard automatically on the screen when an input is necessary.

11.2.2.9 How to make the start of the PC start my application automatically?

To see the option "[automatic start with the PC](#)"^[324] of the runtime.

11.2.2.10 How to make my Optima PLC application communicate with another application?

The problem of the communication with another program (communication inter-processus) can be taken of several different manners.

Communication via a TCP / IP socket

It is the method generally carried for the communication between the Optimalog applications. In Optima PLC, the TCP library provides all necessary functions.

- Advantages
This method functions even though the applications are not on the same PC.
It permits to send important volumes of data.
- Inconveniences
It is a little more complex to program, because it is necessary to take into account the establishment of the communication.
It doesn't permit a communication since the tasks real-time.

Communication via the Windows register

This method consists in using the functions of the WinRegistry library to read and to write data in the Windows register.

- Advantages
Very simple method to put in work.
- Inconveniences
The programs must absolutely function on the same PC.
Doesn't permit to exchange big volumes of data.
It doesn't permit a communication since the real-time tasks.

Communication by shared data area

This method is based on the possibility to share a same area of data memory of the PC between several Windows-based applications. It uses the SharedMemory library.

- Advantages
 - Ease of use in Optima PLC
 - Area data can be shared fairly large (but not to exceed Mo).
 - Instant Communication
 - Once the initialization of the area carried out, access to data can be done from the real-time tasks.
- Inconveniences
 - The programs must absolutely run on the same PC.
 - The applications don't always have the possibility to use this technique.
 - Attention to the consistency of data.

11.2.2.11 How to make display fault messages simply ?

To get an alarm display in a very simple manner, use the predefined flags of A alarms :

- define a mnemonic and a commentary for every variable used, describing each a defect.
- In the program, put to 1 or 0 the variable of defect
- configure the display of the alarms in the runtime.

Other solution for a more elaborated treatment of the alarms: use the software Optim'Alarm, especially conceived in this goal and perfectly integrated to Optima PLC.

11.2.3 Resolution of problems

11.2.3.1 I define some mnemonics on variables standards but they don't appear

To be taken correctly in account in the viewing screens, the definitions of variables standards with mnemonics (registers, inputs / outputs, etc...) must be performed in a block of global variables.

Besides, with a version previous to the v2.9.1, at least one of the declarations made in this block must be used in the rest of the project, otherwise the block is not integrated entirely to the compiled application. From the v2.9.1, the declarations of global variables are integrated systematically to the project.

11.2.3.2 I have a compiling error on a function name described in the documentation

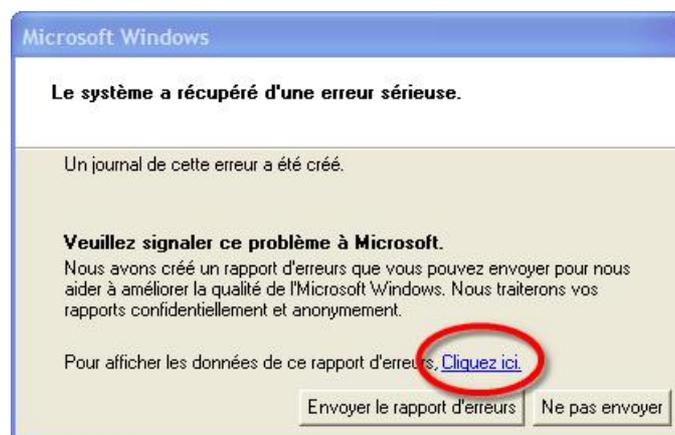
You found in the documentation a function (or a block of function) that you used in your program. But when compiling, a message indicates that the function is unknown.

The function is probably in a library that is not currently loaded. Only POU's defined in loaded libraries are really available or programming. Click on the tab "Libraries" of the POU's tree, and verify the presence of this library. So need, click on "Other available library" and load the one containing the wished function.

11.2.3.3 A blue screen is displayed or the PC restarts during the execution of the runtime.

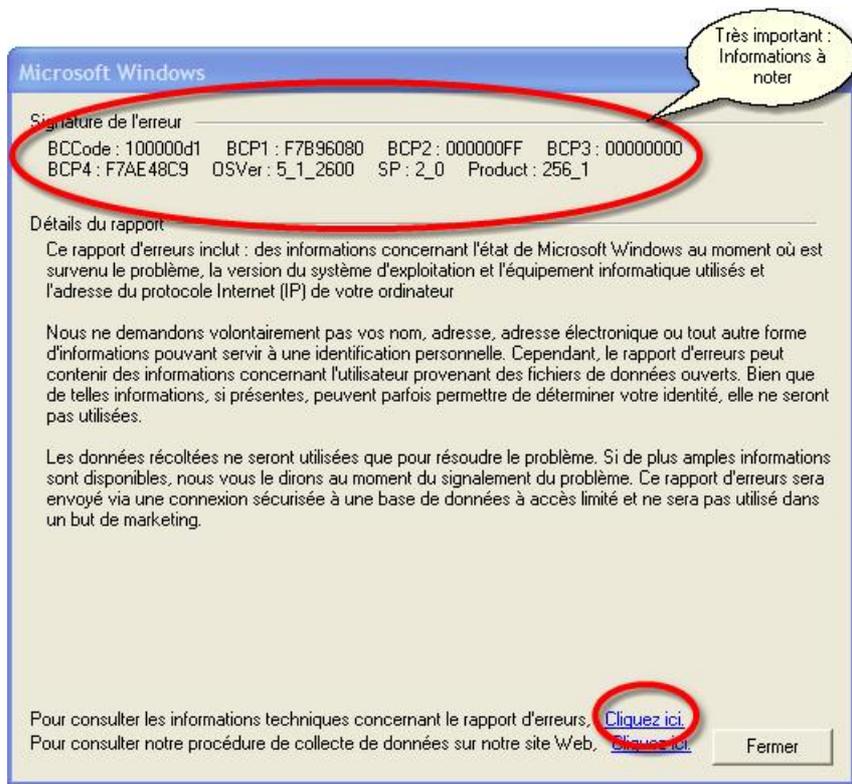
If a blue screen is displayed, note the main information in top of the screen.

In any case, at the time of the start according to the PC, an information message is displayed, asking for confirmation before sending the crash report to Microsoft.



Don't send the report of mistake, save the files mentioned in this crash report, and send them, as well as your project, to OptimaLog for analysis and correction of the problem.

Attention: the backup of the files mentioned below must be made before clicking on not to send."



Files to recuperate and to send to OptimaLog (before closing the first window) :



11.2.3.4 I have a problem with Optima PLC and I don't find the solution in the documentation, who may I contact ?

For the supplementary support, contact Optimalog by email to the following address :

optimaplc@optimalog.com

specifying your coordinates, the precise version of Optima PLC, the nature of your problem, and joining the maximum of files permitting to perform a diagnosis, in particular the files of your application.

Optimalog guarantees a total privacy, no file transmitted will be used outside of the diagnosis.

11.2.4 Various

11.2.4.1 How can I reduce the compilation time of my application?

The compilation time can take, in the extreme cases, a few score of seconds. It doesn't generally exceed about ten seconds.

Besides the speed of the PC used, the factor that influences the more on the compilation time is the number of loaded libraries and therefore of POU's to compile. At the time of the creation of a new project, Optima PLC places by default in the tab Libraries all necessary standards libraries. However, if you don't use some among them, closing them allows to decrease the compilation time considerably.

12 Appendices

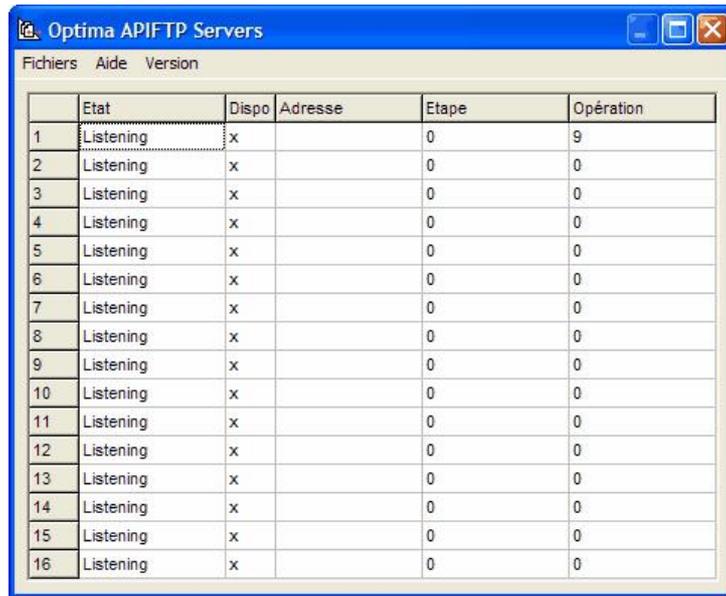
12.1 Optima APIFTP Servers

This software permits to put a handshake of files in place specific in Optima PLC :

- File transfer between PC distant, on local area network or via Internet.
- Communication with stations functioning under Apigraf V10.
- Macros of file transfer integrated in the supervision Optima View.

12.1.1 Main page

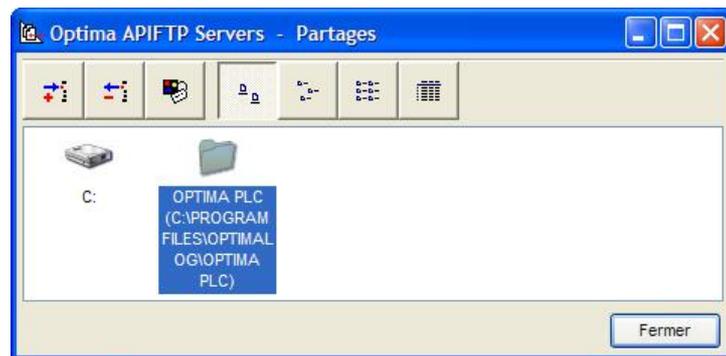
The main page includes a table display of protocol socket servers, allowing to see the activity.



	Etat	Dispo	Adresse	Etape	Opération
1	Listening	x		0	9
2	Listening	x		0	0
3	Listening	x		0	0
4	Listening	x		0	0
5	Listening	x		0	0
6	Listening	x		0	0
7	Listening	x		0	0
8	Listening	x		0	0
9	Listening	x		0	0
10	Listening	x		0	0
11	Listening	x		0	0
12	Listening	x		0	0
13	Listening	x		0	0
14	Listening	x		0	0
15	Listening	x		0	0
16	Listening	x		0	0

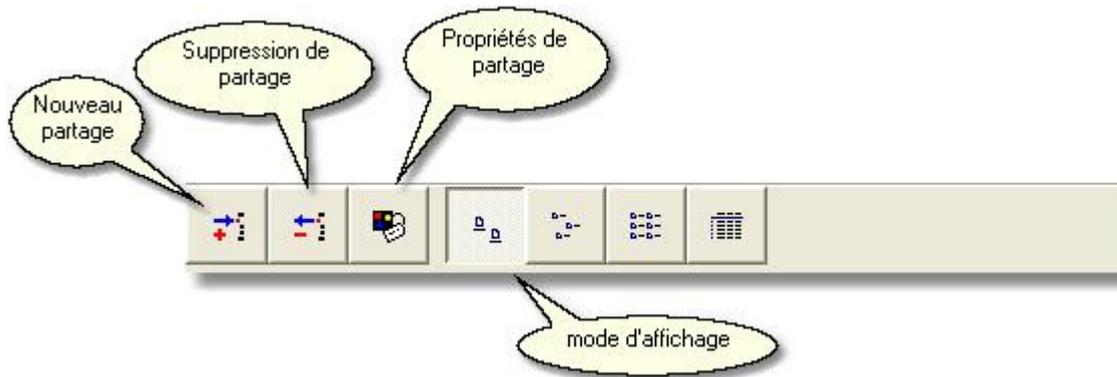
12.1.2 definition of the sharings

The option Sharings of the menu shows the definition window of the sharings.

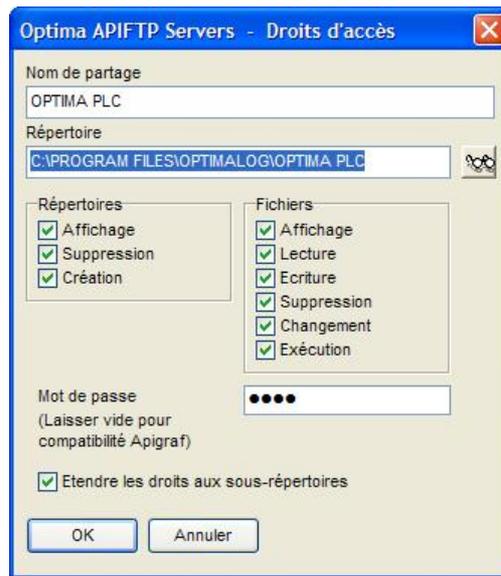


This window permits to add, to modify or to suppress the sharings.

A sharing is the definition of an access right to a directory of the PC.



The options of the sharings are defined by means of the following window :



Several sharings can be defined for the same directory with different rights and passwords, by giving them a different sharing name.

12.2 Codes of APIFTP result

12.2.1 opInProgress

Transaction in progress
Value: -1

The last asked transaction is currently under execution.

12.2.2 opOK

Transaction finished with success
Value: 0

12.2.3 opProtocolError

Error of protocol
Value: 1

12.2.4 opConnectionError

Impossible connection
Value: 10

The distant system doesn't answer. It can be about several problems :

- Routing: the demand doesn't arrive to destination (to verify with a command "ping" if the destination can be achieved).
- Stall by a firewall: the distant system doesn't allow the opening of the necessary service (TCP port number 10260).

- The distant service is not started: under Windows, to verify that Optima APIFTP Servers is started; under Apigraf, to verify that InitAPI is started, and that the services network standards is installed.
- The distant service is saturated, all sockets is already used.

12.2.5 opConnectionAborted

Connection interrupted
Value: 11

12.2.6 opConnectionTimeOut

Connection finished on time out
Value: 12

12.2.7 opNoRessource

Not enough hardware resources
Value: 13

12.2.8 opDestinationUnreachable

No resource permits to reach the destination
Value: 14

The asked distant service is on an IP address that the station doesn't know how to achieve.

12.2.9 opVersionAnswerTimeOut

No receive of number of protocol version
Value: 15

12.2.10 opSourceNotFound

Source file does not exist.
Value : 20

12.2.11 opUnableToOpenSource

Impossible to open source file
Value: 21

12.2.12 opUnableToReadSource

Impossible to read source file
Value: 22

12.2.13 opExistFile

Existing file
Value: 23

Positive answer at the time of a test of file existence.

12.2.14 opPathNotFound

Inexistent path
Value: 24

12.2.15 opAccessDenied

Access Denied
Value: 25

This error indicates a denied access to a file or a directory, because the information of sharing don't permit this access, or because the necessary password is not provided or is incorrect.

With APIFTP Servers, to verify that the concerned directory is defined correctly in the sharings.

12.2.16 opUnableToCreateDest

Creation of the destination file impossible
Value: 30

12.2.17 opUnableToWriteDest

Creating or writing of the destination file impossible
Value: 31

12.2.18 opFileSizeError

Error on the size of file

Value : 40

12.2.19 opUnableToRenameDest

Impossible to rename the destination

Value: 50

12.2.20 opSourceIsDir

The source is a directory

Value : 60

12.2.21 opSourceIsVolumeID

The source is an identifier of volume

Value: 61

12.2.22 opSourceIsWriteProtected

The source cannot be written because it is write-protected.

Value: 62

12.2.23 opUnableToDeleteSource

Impossible to suppress the source

Value: 63

12.2.24 opUnableToRenameSource

Impossible to rename the source

Value: 70

12.2.25 opUnableToCreateDir

Impossible to create the directory

Value: 80

12.2.26 opUnableToDeleteDir

Impossible to suppress the directory

Value: 81

12.2.27 opDirAlreadyExists

The directory already exists

Value: 82

An attempt has been made to create a directory already existing.

12.2.28 opUnableToGetVersion

Impossible to get the version of the file

Value: 90

12.2.29 opUnknownParameter

Unknown parameter

Value: 91

12.2.30 opUnableToExecFile

Impossible to execute the file

Value: 92

The distant system is not able to execute the given file.

12.2.31 opProtocolVersionError

Non available function with the version of the protocol

Value: 199

The distant service is of a former version that doesn't allow the asked function.

12.2.32 opError

Error general APIFTP
Value: 200

A non listed error occurred during the last transaction.

Index

- A -

About Optima PLC Workshop 30
 AllLinksOK function: 310
 ANI 350
 Animate bitmap: 197
 Animation: 190
 ANO 350
 Apigraf 443
 ARRAY 350
 Aspect :: 95
 AT 353
 Attributes of variables 353
 Authorization of modification of variables PLCs via a DDE tie: 308

- B -

Beaches of values 350
 BEEP (Macro PLC): 298
 Bet forward plane or behind plane of an object: 105
 Billposters 7 segments: 279
 Bits string 348
 Block of program 34
 BOOL 348
 Bottom: 96
 BOUTON MARCHE / ARRÊT: 223
 Bouton simple: 114
 BOUTON TYPE ARRÊT D'URGENCE: 225
 Button API: 117
 BUTTON MARKS STOP OF EMERGENCY: 225
 BUTTON WORKS / STOP: 223
 BYTE 348

- C -

C 350
 Calculator 20
 Calendar / Clock 21
 Call of a supervisory form since another page: 286
 Call of a supervisory form with passage of parameters: 287
 Camembert: 187
 Change of language: 306
 Change of size of an object or a group of objects with the help of the function zoom: 104
 CHAR 348
 Characters string 347
 ChnMdp (local Macro): 291
 ChnUserLanguage (local Macro): 291
 Close (Macro locale): 292
 Closing of one page since a button: 287

COLUMN 3 COLORS: 235
 Confirmation 375
 CONSTANT 353
 Constants
 characteristic 355
 characters strings 354
 dates 355
 lasted 355
 numeric 354
 Constants system 24
 Consultation of the DLL user: 305
 Contained of a library 32
 Contained of a project 2
 Contained of the POU's 36, 38, 40, 42, 43, 45
 Contextual menus of a diagram function blocks 55
 Contextual menus of the diagrams ladder 53
 Contextual menus of the SFC diagrams 59
 Copy of objects with the help of the mouse: 105
 Creation of a DELPHI5 program using the USERAPIPLC packet: 309
 Creation of a new file of language: 306
 Creation of a supervisory form: 94
 Creation of an action 60
 Creation of an object of supervisory: 102
 Creation of diagrams Function block 54
 Creation of Ladder diagrams 48
 Creation of SFC diagrams 58
 Cursor: 216

- D -

Date and time 347
 DAY 352
 DAYOFWEEK 352
 Declarations of variables 352
 DECR ###(Macro PLC): 298
 Definition of a group: 314
 Definition of an item: 314
 Definition of the data in read/write: 314
 Definition of the number of local variables: 312
 Deletion of a DLL user of a project: 305
 Deletion of objects of supervisory: 104
 Deletion of objects personalized: 288
 Demand of confirmation 375
 Derivative types 349
 Description of the local variables: 312
 Development of a DLL User: 299
 Développement d'une DLL Utilisateur: 299
 DINT 346
 Displacement and change of size of an object of supervisory: 103
 Display of information 375
 Drivers menu 333
 DT 350
 DWORD 348

- E -

ELSE 358
 ELSIF 358
 END_CASE 358
 END_FOR 358
 END_IF 358
 END_REPEAT 358
 END_STRUCT 349
 END_TYPE 349
 END_WHILE 358
 Error message 375
 Example of creation of a SFC diagram 73
 Example of diagrams function blocks 57
 Example of DLL User: 303
 Example of translation: 306
 Example: Creation of a DDE tie with MICROSOFT® Excel: 308
 ExecMacro procedure: 310
 ExecProg (local Macro): 293
 Execution 20
 Execution of one page since the workshop: 307
 Execution of one page with his/her/its icon: 307
 Exemple de création d'un diagramme ladder 54
 EXIT 358

- F -

Fashion video 20
 File manager 15
 Files create in a project of supervisory: 307
 First utilisation création of a project 8
 Floodgate: 245
 Folder: 155
 Following document 20
 Format Dates / Hour 21
 Function 34
 Function block 34

- G -

General compilation 23
 General principles: 92, 308, 312
 Generalities: 299, 312
 Generic types 348
 Global variables 33
 Grafcet 58
 Graph: 172
 Grid: 97

- H -

HEART 358
 Help 30

HIS/HER/ITS 350
 Histogram: 183
 Horodatage 347
 HOURS 352

- I -

I 350
 Import of objects personalized: 289
 Impression 79
 INCR ###(Macro PLC): 298
 Initialization of the types and variable 356
 Initialization of types and variable multi-elements 356
 Initialization of variables 356
 Insertion pictures succesives: 193
 Installation of Optima PLC 4
 Installation of the application: 315
 INT 346
 INVSA ###(Macro PLC): 298
 IsLinkedToAPI function: 309

- L -

L 350
 Language programming 11
 Language software 11
 Lasted 347
 LED: 237
 LEVER WITHOUT COLOR: 225
 LEVIER SANS COULEUR: 225
 Libraries "Project" 36
 Line: 169
 LinkToAPI function: 309
 LINT 346
 List of the LOUSE 14
 List of the windows 21
 LREAL 346
 LWORD 348

- M -

Macros: 101
 Management of contacts 49
 Management of the passwords: 287
 Management of the time 374
 Material interfaces 24
 Menu Affichage - Barres d'outils: 83
 Menu Control 324
 Menu Create Shortcuts: 80
 Menu Display - Bar of tools: 83
 Menu Display - Window Function: 83
 Menu edition - All to select: 80
 Menu edition - to Annul: 80
 Menu edition - to Copy: 80
 Menu edition - to Glue: 80

Menu edition - to Suppress: 80
 Menu Exploring Apigraf IP: 83
 Menu Expressions 330
 Menu file - New: 80
 Menu file - Quit: 80
 Menu file - Record as: 80
 Menu file - Save: 80
 Menu file - Select project: 80
 Menu file - to Open: 80
 Menu Macros - local Macros: 86
 Menu Macros - Macros PLC: 86
 Menu Macros - Macros User: 86
 Menu Object - to Add: 84
 Menu Object - to Import: 85
 Menu Object - to Suppress: 84
 Menu of management of the libraries 46
 Menu of management of the POU's 47
 Menu Options - Edition configuration: 90
 Menu Options - general Options: 89
 Menu Outputs Text 331
 Menu PLC 327
 Menu Translations user - Extraction Auto texts of the page: 81
 Menu Translations user - New file language: 81
 Menu Translations user - to Open a file language: 81
 Menu Variables 329
 MINUTES 352
 MONTH 352

- N -

Name of the OPC server: 314
 Negation 51

- O -

OF 350
 Oh 350
 --OLD KEYWORDS-- 83, 98, 99, 114, 223, 225, 226, 230, 299
 OLE Container: 281
 Options Publisher 21, 22
 Options: 99
 Output text files 25, 376

- P -

Panel: 160
 Picture: 199
 PLCState function: 311
 Police: 96
 Position: 98
 Previous document 20
 PrintPage (local Macro): 294
 Processes of POU's 353
 Programming 345

Advanced 374
 Apigraf language 362
 Diagrams Function block 54
 Ladder diagrams 48
 Language HIM 359
 SFC (Grafcet) diagrams 58
 ST. language 357
 Publisher Optima PLC 14

- Q -

Qualifier of action 61
 D 63
 DS 65
 L 62
 N 61
 P 63
 P1 and P0 64
 S and R 62
 SD 64
 SL 66

- R -

R 350
 Radio group: 110
 REAL 346
 real-time kernel 441
 Recipe: 276
 Registration of objects Personalized: 288
 REPEAT 358
 RETAIN 353
 RETURN 358
 Rod of progression: 205
 Rod of tool of a diagram Function blocks 55
 Rod of tool of a SFC diagram 59
 Rod of tool of management of the display of the windows 31
 Rod of tool of the diagrams ladder 49
 RR 350
 RunHelp (local Macro): 294
 RunHelpIndex (local Macro): 294
 RunHelpTopic (local Macro): 294

- S -

Samples tables 376
 Scale: 186
 Scroll bar: 221
 Scrolling: 157
 SECOND 352
 SEEING DOUBLE: 233
 SEEING SIMPLE: 231
 Seen meter to needle: 213
 Seen meter: 203
 Seized of the macros for the objects: 105

SELECTEUR 2 / 3 POSITIONS: 226
 SELECTEUR ARRÊT / AUTO / MANU: 230
 Selection of a project: 317
 Selection of an object or a group of objects of
 supervisory: 103
 Selection project 13
 SELECTOR 2 / 3 POSITIONS: 226
 SELECTOR STOP / AUTO / MANU: 230
 ServApiOPC application: 313
 SET. = (Macro PLC): 299
 Shape: 167
 Simple button: 114
 SINT 346
 SLOT 358
 Slot to nick: 108
 Starting of the application: 316
 Statute of the application 320
 STRUCT 349
 Structure of data 349
 Subgroup 34
 SuperG (local Macro): 295

- T -

Tables of data 350
 Task 32
 Techniques of management of the time 374
 Text parading: 196
 Text: 106
 THEN 358
 Time 347, 374
 To 350
 To create a POU 47
 To leave 15
 Tonnage: 201
 Total compilation 18
 TYPE 349
 Types 33
 Types enumerated 349

- U -

UDINT 346
 UINT 346
 ULINT 346
 UnLinkToAPI function: 310
 UNTIL 358
 Use of a DLL User in a project: 305
 Use of the language files in the supervisory forms: 306
 USINT 346

- V -

Vanne1: 247
 VAR 352
 VAR_EXTERNAL 352

VAR_GLOBAL 352
 VAR_IN_OUT 352
 VAR_INPUT 352
 VAR_OUTPUT 352
 VAR_TEMP 352
 Vat: 242
 Versions 431
 Viewer HTML: 159
 Visualization of the local variables: 312
 VisuText (local Macro): 295

- W -

WEEKNB 352
 WFCopy (local Macro): 295
 WFCopyDirectory (local Macro): 296
 WFDelete (local Macro): 296
 WFDeleteDirectory (local Macro): 296
 WFRename (local Macro): 296
 WFSaveFileAttr (local Macro): 296
 WHILE 358
 Whole 346
 Window of edition with menu unwinding: 151
 Window of edition: 139
 Window system 15
 WNetSChangeFileName (local Macro): 296
 WNetSCopyFileRemote (local Macro): 297
 WNetSCreateDirectory (local Macro): 297
 WNetSDeleteDirectory (local Macro): 297
 WNetSReadAndMoveFile (local Macro): 297
 WNetSReadFile (local Macro): 297
 WNetSSetFileAttr (local Macro): 298
 WNetSWriteAndMoveFile (local Macro): 297
 WNetSWriteFile (local Macro): 298
 WORD 348

- Y -

YEAR 352
 YEW 358