1. OVERVIEW

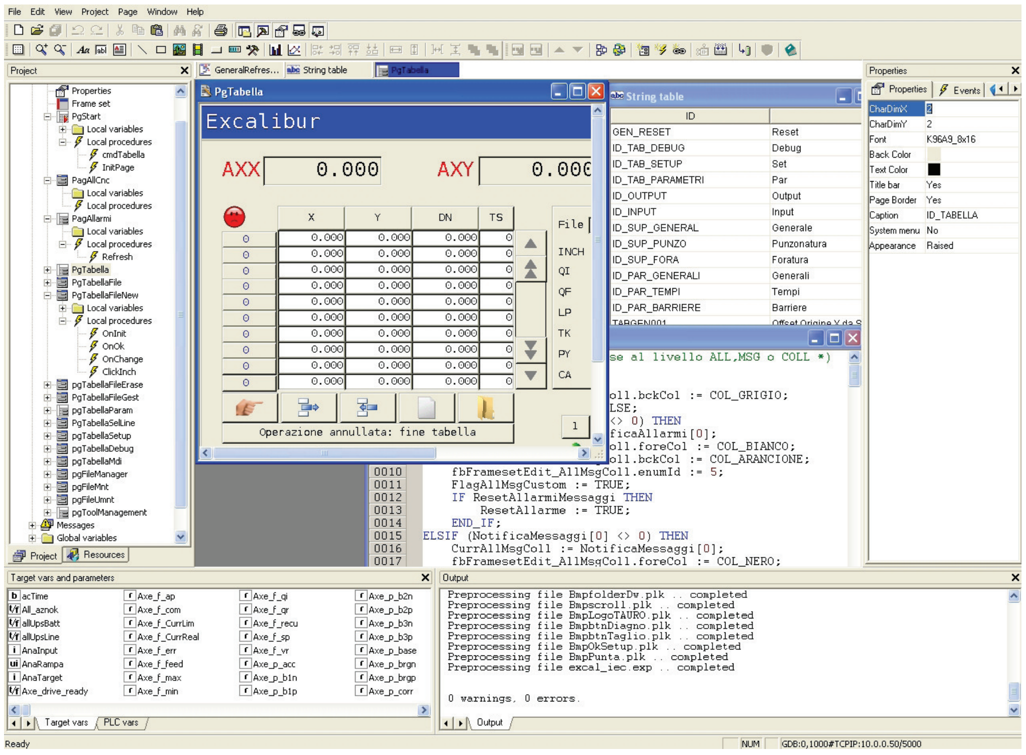
PageLab is a software application that allows the developer to create user interfaces for embedded systems based on HMI runtime.

PageLab is an easy to learn and use software, which allows the user to implement graphical interfaces in a visual way. The realized pages are viewed in PageLab as they will appear on the final target.

Thanks to its multi-pages structure, PageLab can support HMI (Human Machine Interface) applications with an arbitrary number of pages.

It is equipped with a considerable number of tools to realize even complex applications and it interfaces directly to the PLC IEC1131 LogicLab compiler for managing the variables which are defined in the target PLC application.

The following paragraphs show you the main features of this product.



String table

ID	
GEN_RESET	Reset
ID_TAB_DEBUG	Debug
ID_TAB_SETUP	Set
ID_TAB_PARAMETRI	Par
ID_OUTPUT	Output
ID_INPUT	Input
ID_SUP_GENERAL	Generale
ID_SUP_PUNZO	Punzonatura
ID_SUP_FORA	Foratura
ID_PAR_GENERALI	Generali
ID_PAR_TEMPI	Tempi
ID_PAR_BARRIERE	Barriere

Properties

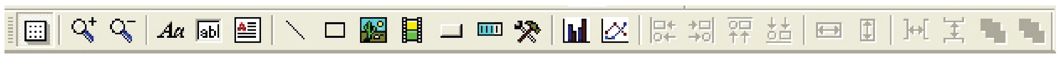
CharDimX	2
CharDimY	2
Font	K96A9_8x16
Back Color	
Text Color	■
Title bar	Yes
Page Border	Yes
Caption	ID_TABELLA
System menu	No
Appearance	Raised

```
    fbFramesetEdit_AllMsgColl.enumId := 5;
    FlagAllMsgCustom := TRUE;
    IF ResetAllarmiMessaggi THEN
        ResetAllarme := TRUE;
    END_IF;
ELSIF (NotificaMessaggi[0] <> 0) THEN
    CurrAllMsgColl := NotificaMessaggi[0];
    fbFramesetEdit_AllMsgColl.foreCol := COL_NERO;
```

```
Preprocessing file BmpfolderDw.plk .. completed
Preprocessing file Bmpscroll.plk .. completed
Preprocessing file BmpLogoTAURO.plk .. completed
Preprocessing file BmpbtnDiagno.plk .. completed
Preprocessing file BmpbtnTaglio.plk .. completed
Preprocessing file BmpOkSetup.plk .. completed
Preprocessing file BmpPunta.plk .. completed
Preprocessing file excal_iec.exp .. completed


0 warnings, 0 errors.
```

1.1 MAIN ELEMENTS

Set of controls

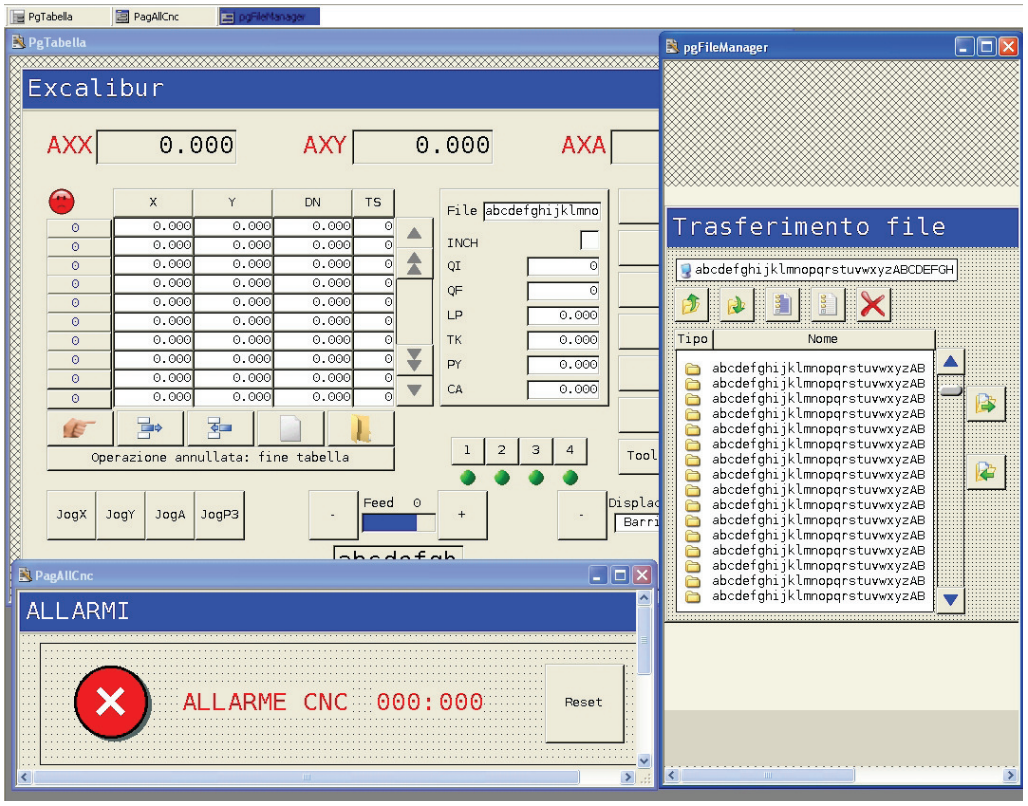


Each page may contain an arbitrary number of defined graphic controls. There are two classes of graphic controls:

- Static controls: drawing tools such as lines, rectangles, and figures.
- Dynamic controls: multilayered objects, which enable data and images display and user interaction (strings, editboxes, textboxes, buttons, progress, charts and trends, custom controls).

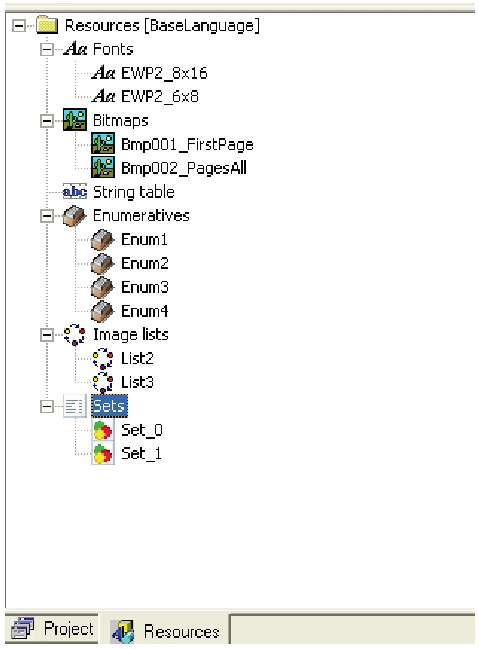
PageLab is an open system, allowing the implementation of custom controls which may be included in the target system.

Multi-pages structure



PageLab supports the definition of an arbitrary number of pages (full-screen or pop-up). Each page may contain links to other pages, so that the whole project takes a tree structure.

Resources management

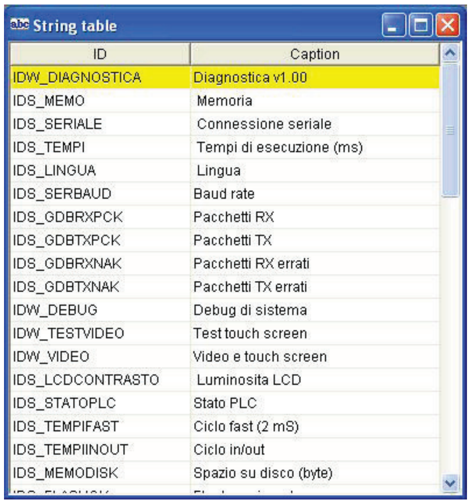


The controls' properties in the page are not statically defined in the project code, but they can be managed separately as resources.

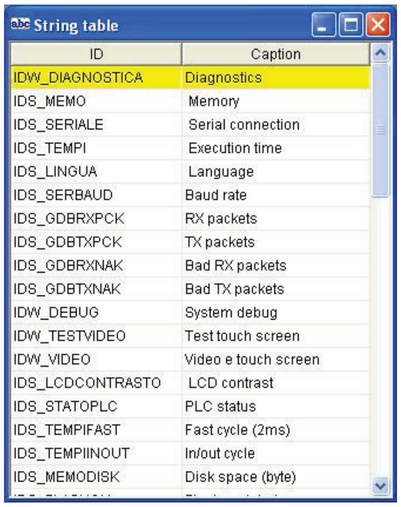
Resources include fonts for characters display, images, string table, enumerated data types, and elements sets.

Specifically regarding the images, PageLab allows to import bitmap files directly from the Windows-formatted file (*.bmp*, *.gif*, *.emf*, *.jpg*, *.ico* etc.).

Languages management



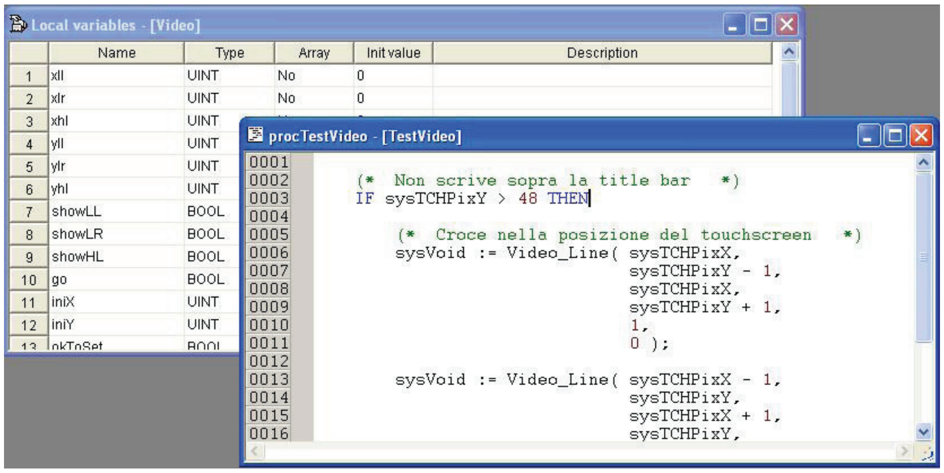
ID	Caption
IDW_DIAGNOSTICA	Diagnostica v1.00
IDS_MEMO	Memoria
IDS_SERIALE	Connessione seriale
IDS_TEMPI	Tempi di esecuzione (ms)
IDS_LINGUA	Lingua
IDS_SERBAUD	Baud rate
IDS_GDBRXPCK	Pacchetti RX
IDS_GDBTXPCK	Pacchetti TX
IDS_GDBRXNAK	Pacchetti RX errati
IDS_GDBTXNAK	Pacchetti TX errati
IDW_DEBUG	Debug di sistema
IDW_TESTVIDEO	Test touch screen
IDW_VIDEO	Video e touch screen
IDS_LCDCONTRASTO	Luminosita LCD
IDS_STATOPLC	Stato PLC
IDS_TEMPIFAST	Ciclo fast (2 mS)
IDS_TEMPIINOUT	Ciclo in/out
IDS_MEMODISK	Spazio su disco (byte)



ID	Caption
IDW_DIAGNOSTICA	Diagnostics
IDS_MEMO	Memory
IDS_SERIALE	Serial connection
IDS_TEMPI	Execution time
IDS_LINGUA	Language
IDS_SERBAUD	Baud rate
IDS_GDBRXPCK	RX packets
IDS_GDBTXPCK	TX packets
IDS_GDBRXNAK	Bad RX packets
IDS_GDBTXNAK	Bad TX packets
IDW_DEBUG	System debug
IDW_TESTVIDEO	Test touch screen
IDW_VIDEO	Video e touch screen
IDS_LCDCONTRASTO	LCD contrast
IDS_STATOPLC	PLC status
IDS_TEMPIFAST	Fast cycle (2ms)
IDS_TEMPIINOUT	In/out cycle
IDS_MEMODISK	Disk space (byte)

Strings and enumerated data types are structured as to ease the multilingual device; moreover PageLab provides a function to export/import the above mentioned elements to/from a text file, in order to simplify the translation from a language to another.

Variables and procedures



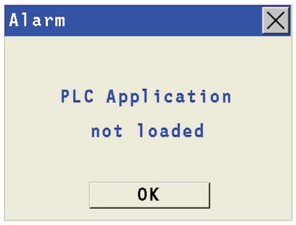
	Name	Type	Array	Init value	Description
1	xll	UINT	No	0	
2	xlr	UINT	No	0	
3	xhl	UINT			
4	yll	UINT			
5	ylr	UINT			
6	yhl	UINT			
7	showLL	BOOL			
8	showLR	BOOL			
9	showHL	BOOL			
10	go	BOOL			
11	iniX	UINT			
12	iniY	UINT			
13	okToSet	BOOL			

```

(*  Non scrive sopra la title bar   *)
IF sysTCHPixY > 48 THEN

    (*  Croce nella posizione del touchscreen   *)
    sysVoid := Video_Line( sysTCHPixX,
                           sysTCHPixY - 1,
                           sysTCHPixX,
                           sysTCHPixY + 1,
                           1,
                           0 );

    sysVoid := Video_Line( sysTCHPixX - 1,
                           sysTCHPixY,
                           sysTCHPixX + 1,
                           sysTCHPixY,
```

PageLab enables the implementation of procedures which may be as complex as you want in the ST language. Through these procedures, the user can interact with the PageLab application, the PLC application or the target system variables to customize the interface's behaviour or the whole CNC.

1.2 RUN-TIME FUNCTIONALITIES

Asynchronous messages management


Alarm

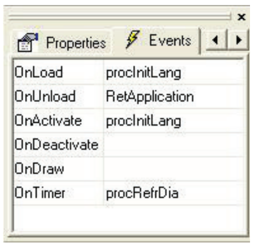
PLC Application
not loaded

PageLab supports the issue of asynchronous messages whatever their complexity. You can entirely customize the issue messages management by typing a simple ST procedure.

Multilingual support

PageLab allows you to change strings, resources, and enumerations language without recompiling nor reloading the application.

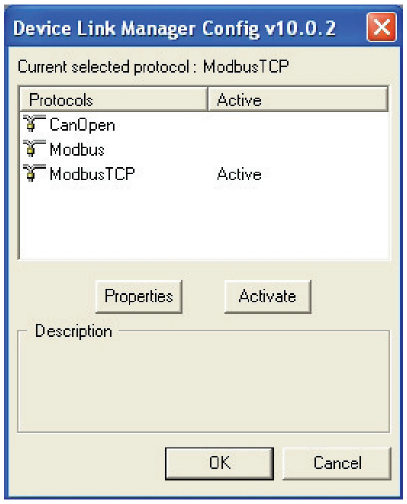
Events management



OnLoad	procInitLang
OnUnload	RetApplication
OnActivate	procInitLang
OnDeactivate	
OnDraw	
OnTimer	procRefrDia

PageLab applications are structured in events; the user may seize the available events and manage them through ST-coded procedures.

1.3 COMMUNICATING WITH THE TARGET


Device Link Manager Config v10.0.2

Current selected protocol : ModbusTCP

Protocols	Active
CanOpen	
Modbus	
ModbusTCP	Active

You can establish the communication with the target device through the PC communication drivers, thus using one of the available custom protocols (which can be easily implemented thanks to the modular structure of the communication system).

2. CREATING A SIMPLE PAGELAB PROJECT

2.1 PURPOSE OF THIS CHAPTER

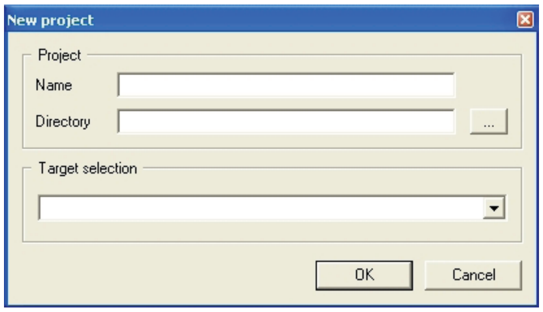
This chapter aims to lead the user to realize a simple HMI project with PageLab, through a sequence of easy steps.

Here below you can find the list of this chapter's topics.

- Creating a new project: starting at zero the realization of a HMI project.
- Inserting the first page in the project.
- Inserting a secondary page.
- Inserting static controls: how to insert simple objects (lines, rectangles, etc.) in a page.
- Inserting static images: how to insert an image in a page, starting at a *.bmp* file.
- Inserting strings: how to insert a text label.
- Inserting edit boxes: how to access the data of the system and the control PLC, how to declare new variables, how to insert text frames to view/edit these data.
- Inserting buttons: learning to use an essential control for the interaction between the user and the system.
- Compiling and downloading the project.

2.2 CREATING A NEW PROJECT

Launch PageLab, then select the *New Project* command from the *File* menu. The following dialog box appears.



Type the name you want to assign to the project in the *Name* field, and in the *Directory* field specify the directory where you want to create the project folder.

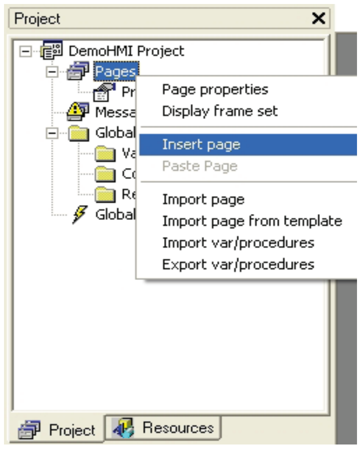
Select the target which will execute the HMI from the *Target selection* menu. The contents of this menu can be customized: if the desired target does not appear in the list, refer to your hardware provider.

Confirm your choice by pressing *OK*. PageLab automatically creates the folder *l:\Demo manuale\Demo HMI* as specified in *Directory*.

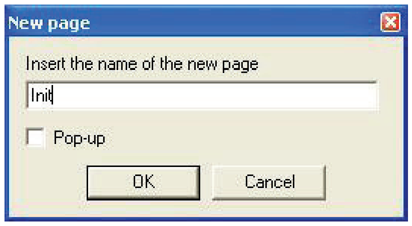
2.3 INSERTING THE FIRST AGE IN THE PROJECT

2.3.1 CREATING A NEW PAGE

To insert a new page in the project, right-click on the *Pages* item of the project tree.

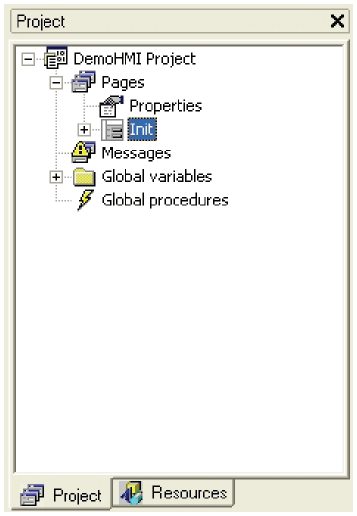


Select the *Insert page* option from the menu which has just shown up. This causes a dialog box to appear where you have to specify the page name and whether the page is a pop-up one or not.

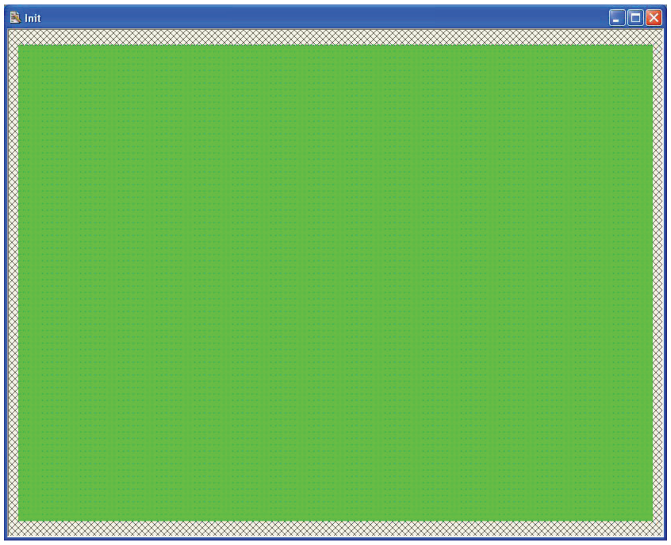


If you do not select the *Pop-up* property when creating it, the page is called *Child Page*. Its main feature is that it fits the whole video area. Consequently the user cannot define position and size of a child page because they are automatically set depending on the video area and on an eventual frame set (see 4.2).

Choose to create a child page and call it *Init*: type the name *Init* in the apposite field and press *OK* to confirm your choice. A new node appears in the pages folder of the project tree.



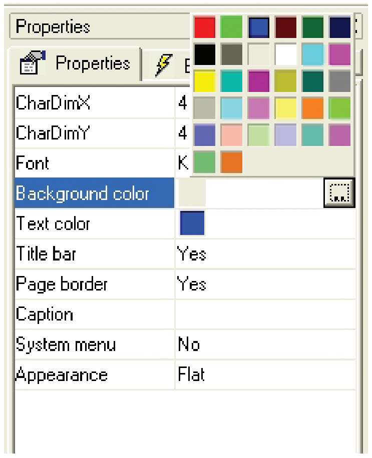
Double-click on the *Init* item to open the document with this page preview, which is blank at the moment.



2.3.2 EDITING THE COLORS OF THE PAGE

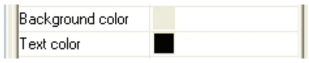
You can edit the background color of the page and the foreground default text color through the page properties: double-click in the *Background Color* field. A little button appears.

Pressing it, the colors palette appears. Then you can select the desired color.



CharDimX	4
CharDimY	4
Font	K
Background color	⬜
Text color	🟦
Title bar	Yes
Page border	Yes
Caption	
System menu	No
Appearance	Flat

Choose grey as background color and black as default text color.

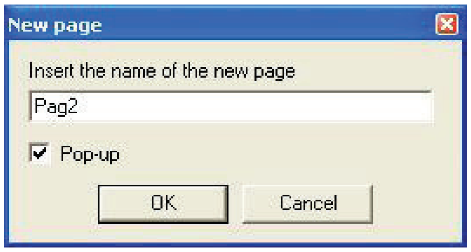


Background color	⬜
Text color	⬛

2.4 INSERTING A SECONDARY PAGE

2.4.1 CREATING A SECONDARY PAGE

Let us assume that you want to create a secondary page: right-click on the *Pages* item of the project tree and choose the *Insert page* option from the contextual menu. Type the name *Pag2* in the dialog box which appears and select the pop-up property.

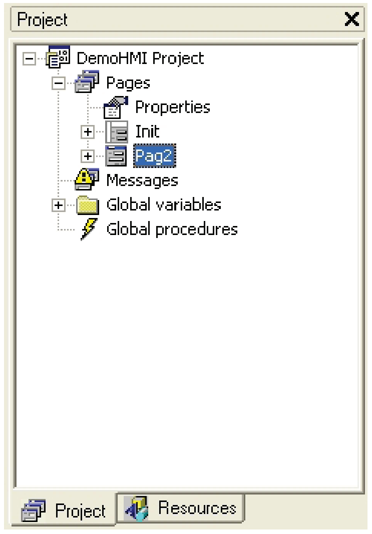

New page

Insert the name of the new page

Pag2

[✓] Pop-up

Consequently a new item appears in the *Pages* folder of the project tree.

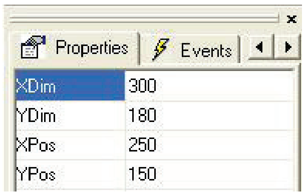


2.4.2 DIMENSIONING AND SETTING THE SECONDARY PAGE

Note that the icon of the *Init* page different from the new *Pag2* one. In fact, the last one has been created as pop-up page, whereas the first one has been created as child page.

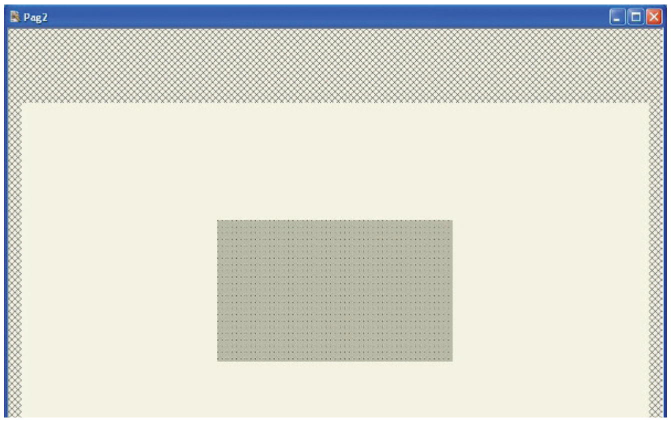
Pop-up pages are not subjected to any restriction from the frame set (see 4.2): their dimensions and positions can be chosen by the user.

Assign to the secondary window the dimensions 300x180 pixel and set it (x, y) = (250, 150) because these are the top left-hand corner's coordinates of the window. Double-click on the *Pag2* item of the project tree. In this way you open the corresponding document. Assign dimensions and position.



XDim	300
YDim	180
XPos	250
YPos	150

After editing the colors, too, the new window will look like the picture below.

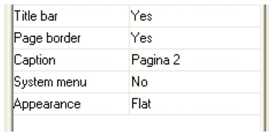


The grey area in the centre is the active area of the *Pag2* page, whereas the clearer area which surrounds it represents the video area of the target system. In this way you obtain a clear vision of the new page placement.

2.4.3 VIEWING THE TITLE BAR AND THE SYSTEM BUTTON

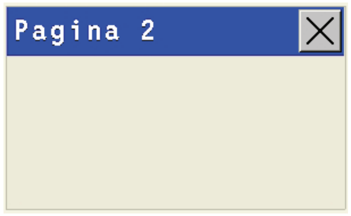
PageLab enables the automatic creation of a title bar (`Title bar` properties = `Yes`) and of a button to close the page (`System menu` properties = `Yes`), besides the print of a text string as title (`Caption` properties).

Let us assume that you want to activate the title bar and the close button, and to print the `Pagina 2` string as title.

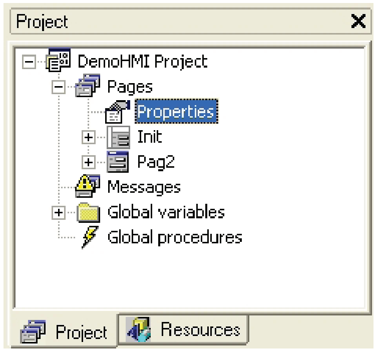


Title bar	Yes
Page border	Yes
Caption	Pagina 2
System menu	No
Appearance	Flat

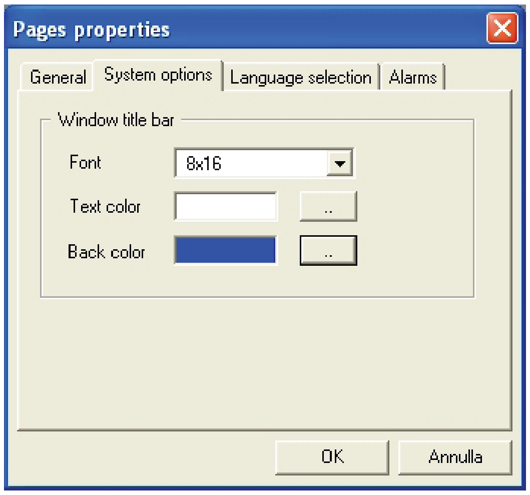
Then the secondary page looks like the following picture.



The text and the background color and the used font are the same for all the pages of the project, so you will not find them in this specific page properties. In order to customize these features, double-click on the `Properties` item of the project tree.

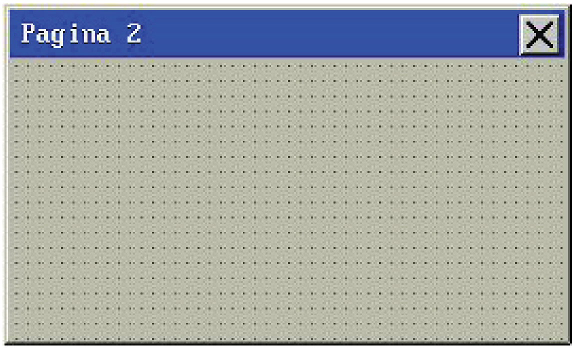


A multi-tabs window opens. In *System options* assign the font (in this case 8x16), the text color and the background color (in this case respectively white and blue).



Font: 8x16
Text color
Back color

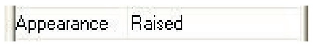
Then the secondary page looks like the following figure.



2.4.4 ASSIGNING A STYLE TO THE WINDOW

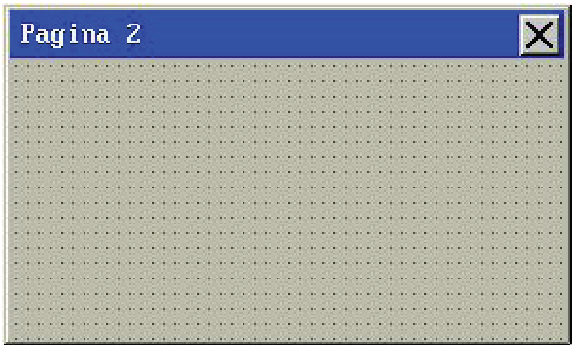
PageLab supports three styles for the windows, which you can select through the *Appear-ance* property: *Flat* (the default style when you create a window), *Sunken* and *Raised*.

Choose the last one.



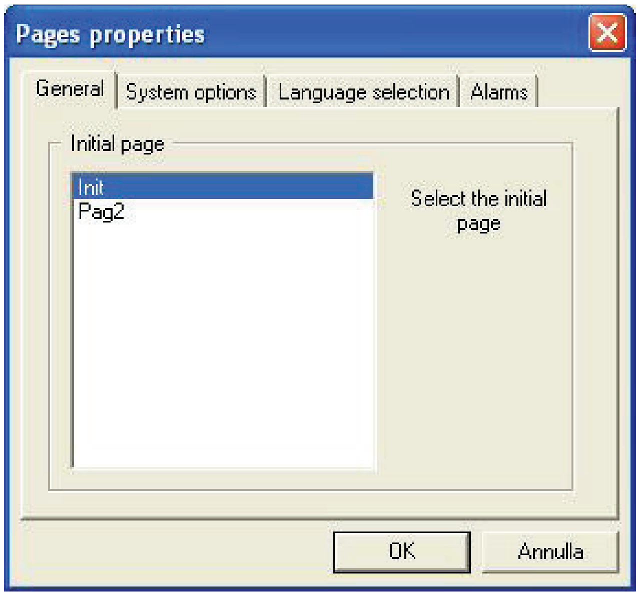
Appearance Raised

The window looks like the picture below.



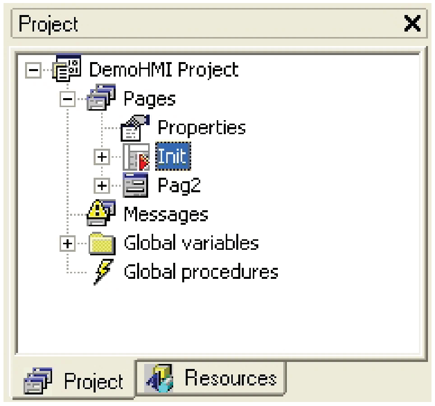
2.4.5 CHOOSING THE START WINDOW

The user has to indicate the start window of the whole HMI project. The start window will open at the HMI application start. If the project consists in one single page, the system will take this one as start page. You can indicate the start page in the project properties window, which you can open by double-clicking on the `Properties` item of the project tree. The `General` window is used for this purpose.



In order to indicate the start page, select the desired one from the list. Then confirm your choice by clicking `OK`.

The start page is marked in the project tree by a red triangle.



2.5 INSERTING STATIC CONTROLS

The two pages which you have just created are blank yet. Go back to the first page (`Init`) and start inserting some controls.

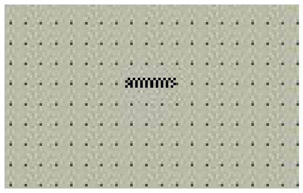
Static controls are objects which are drawn once, when opening the page, and they do not change until the page is active.

2.5.1 INSERTING A LINE

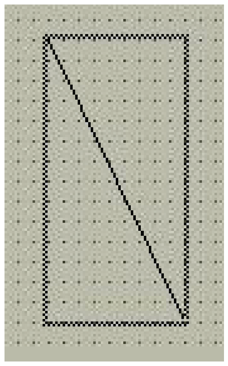
Insert a line by clicking the corresponding button in the `Page toolbar`.

Move the mouse to the active area of the page. A cross + appears. The object will be inserted in the grid near to the mouse cursor.

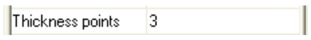
Confirm the insertion point by left-clicking. A new `Line` control appears. It has a default size and horizontal alignment.



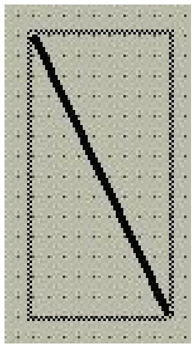
You can resize it by dragging one of the two ends of the line.



You can edit the line thickness through the `Thickness points` property of the control. For example, assign a 3 pixel thickness.


Thickness points 3

In the page preview you can see how the line looks like.

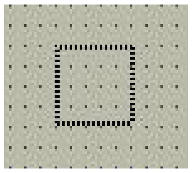


2.5.2 INSERTING A RECTANGLE IN THE PAGE

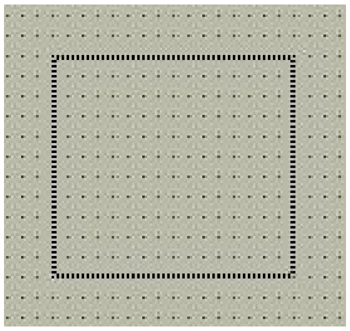
Press the corresponding button in the `Page toolbar`.

Move the mouse to the active area of the page. A cross + appears. The object will be inserted in the grid near to the mouse cursor.

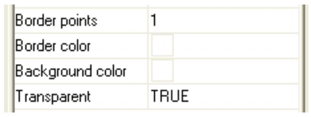
Confirm the insertion point by left-clicking. A new `Rectangle` control appears. It has a default size.



You can edit both the dimensions dragging one of the rectangle vertexes, or one dimension at a time dragging one of the rectangle's sides.

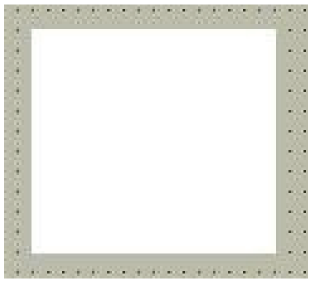


You can customize the border and the background color and the transparency through the control properties. For example, make the rectangle white and opaque with white border and thickness set to 1.

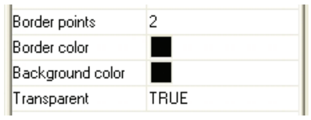


Border points	1
Border color	⬜
Background color	⬜
Transparent	TRUE

In the page preview you can see how the rectangle looks like.

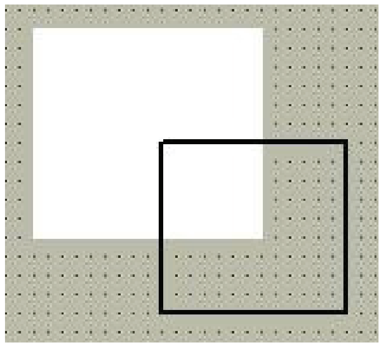


Now superimpose another rectangle to the first one. Let us assume that you want the new rectangle to be transparent with black borders, and thickness set to 2.



Border points	2
Border color	⬛
Background color	⬛
Transparent	TRUE

In the page preview you will see the following image.



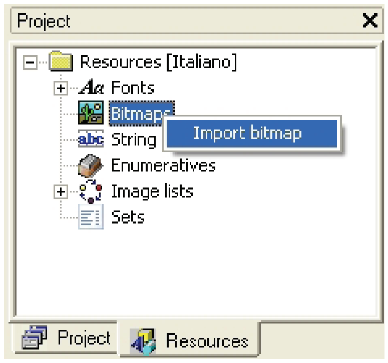
2.6 INSERTING STATIC IMAGES

The following paragraph shows you how to insert static images in the page. Static images are different from animations (images which may change dynamically, even though they have fixed position and dimensions) and from floating images (images which move in the page).

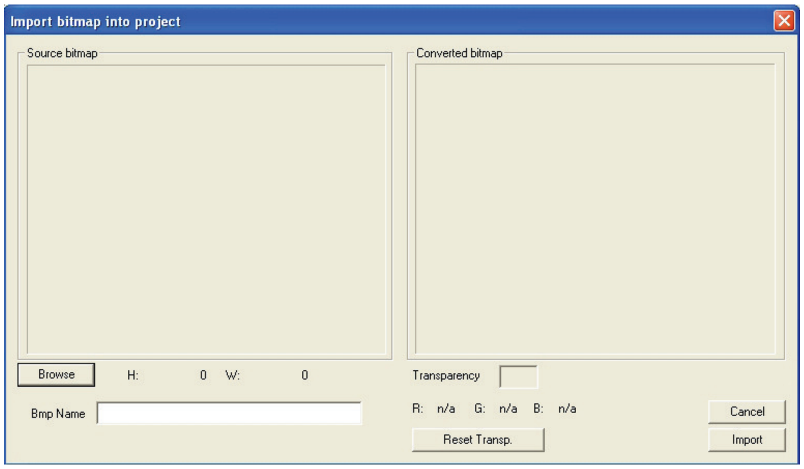
2.6.1 IMPORTING A BITMAP IN THE PROJECT

Image that has to be visualized must be available on PC as a basic Windows image file (*.bmp*, *.dib*, *.emf*, *.gif*, *.ico*, *.jpg*, *.wmf* ...). If this pre-condition holds, you can start the importing procedure.

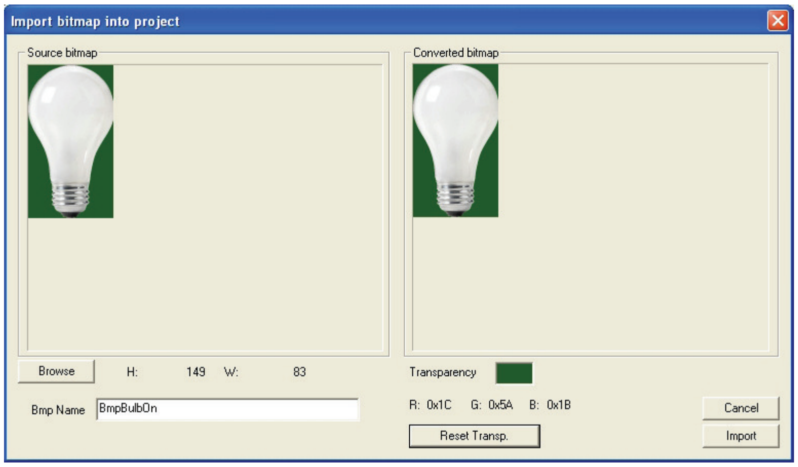
Right-click the *Bitmaps* item in the resources tree and select the *Import bitmap* command in the contextual menu which appears.



A dialog window opens.



Pressing the *Browse* button, you can navigate in the computer resources and select the source file. In this case, the source file is *BulbOn.jpg*, which represents a lighted bulb.


Import bitmap into project

Source bitmap

Converted bitmap

Browse H: 149 W: 83

Transparency

Bmp Name: BmpBulbOn

R: 0x1C G: 0x5A B: 0x1B

Cancel

Reset Transp.

Import

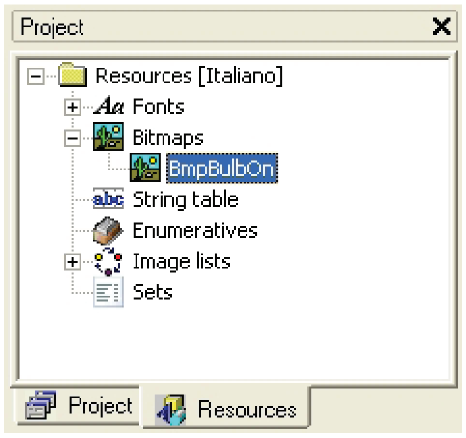
In the *Bitmap Name* field, you can assign the bitmap name which will appear in the resources tree; the default name is the file name without extension and preceded by the *Bmp* prefix.

The *Transparency color* field lets you specify the transparency color, that is a color which will not be really drawn but will let the elements appear through the bitmap background.

You can customize the transparency color by taking the desired one with the mouse from the *Converted bitmap* window.

RGB indicate the transparency color components. If the values are *n/a* it means that no transparency color has been selected. The *Reset Transp.* button lets to cancel the last selected transparency color.

At last you can confirm the operation by clicking the *Import* button. The imported bitmap appears as a new item in the resources tree.

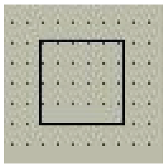


2.6.2 ASSOCIATING AN IMPORTED BITMAP WITH AN IMAGE CONTROL

The control which is aimed to display the static images is called *Image*: press the corresponding button in the *Page toolbar*.

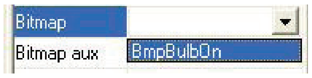
Move the mouse to the active area of the page. A cross + appears. The object will be inserted in the grid near to the mouse cursor.

Confirm the insertion point by left-clicking. A new blank frame appears.

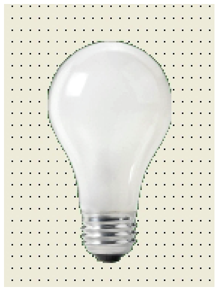


Trough the `Bitmap` property specify the image which this `Image` control must display.

Choose the desired bitmap from the list; in this case, you can see and select the only bitmap which you have imported: `BmpBulbOn`.



The control changes its size to be compatible with the assigned bitmap measures. The image in the page preview looks like the following picture.



2.7 TEXT STRINGS

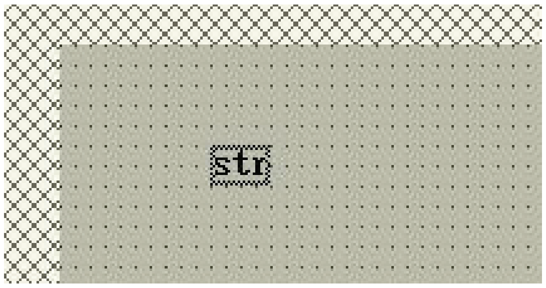
Text strings are not part of static controls because they have some properties which let them change in a page through time. Visibility, selection, and refresh may be assigned to variables, which may change their value at any time.

2.7.1 INSERTING A TEXT STRING

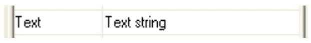
Click the corresponding button in the `Page toolbar`.

Move the mouse to the active area of the page. A cross + appears. The object will be inserted in the grid near to the mouse cursor.

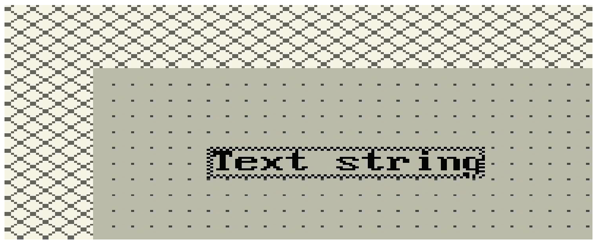
Confirm the insertion point by left clicking. A new `Static` (that is string) control with the default text `str` appears.



You can edit the contents of the string through the `Text` property of the control. For example, `Text string`.


Text Text string

The page preview looks like the image below.



This is the basic use of the string. Alternatively you can assign strings by taking them from the resources (see 4.9.3).

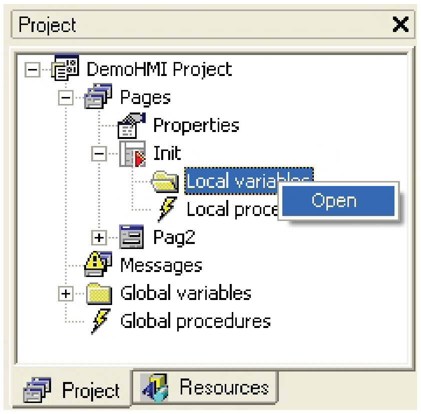
2.8 DATA MANAGEMENT IN PAGELAB

This paragraph shows you the variables management in PageLab. It is possible to distinguish the data in local variables (visible in the page scope only) and global variables (visible from every page). For some controls it is possible to use parameters and sets.

2.8.1 DECLARING A LOCAL VARIABLE

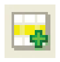
First of all declare a local variable, which you can use just in the specific page where the declaration takes place.

In the pages tree, under the `Init` page item, right-click on the `Local variables` item and select `Open` in the contextual menu which appears.

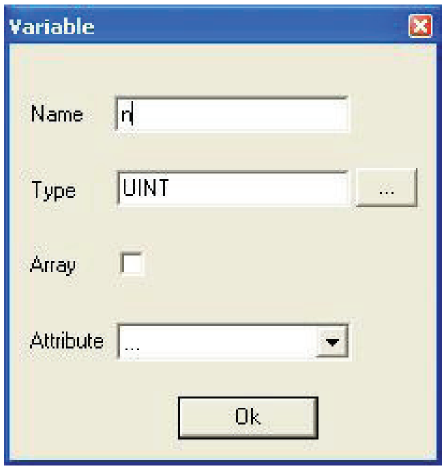


The local variables editor window opens. It is blank at present.

Click the `New record` button in the `Project toolbar.`



A dialog window opens requesting to specify the new variable's basic features. We can declare *n* as a new 16 bit unsigned integer variable.


Variable

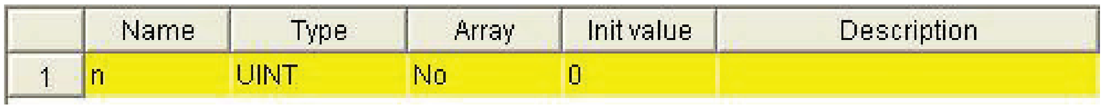
Name: n

Type: UINT

Array: ☐

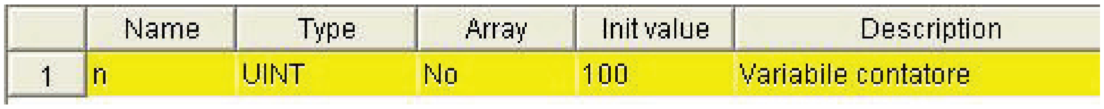
Attribute: ...

Confirm the operation by clicking *Ok*. The new corresponding record is added to the variables editor.



	Name	Type	Array	Init value	Description
1	n	UINT	No	0	

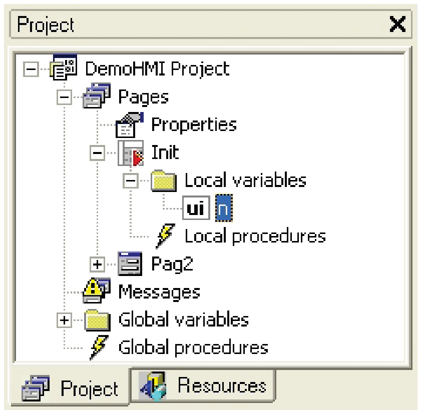
You can change this new variable's features editing the fields of the record which you have just created. For example, you may assign an initial value different from null and a comment.



	Name	Type	Array	Init value	Description
1	n	UINT	No	100	Variabile contatore

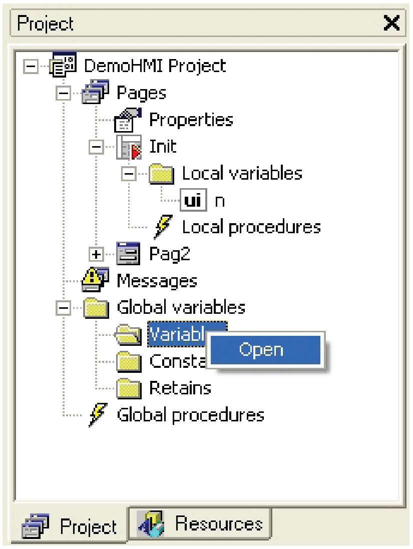
When you save the project by clicking the apposite button

or when you close the variables editor, PageLab adds a new item in the pages tree. It corresponds to the local variable which you have just declared.



2.8.2 DECLARING A GLOBAL VARIABLE

Let us assume that you want to declare a floating point global variable *t*: right-click on the *Variables* item under the *Global variables* node of the resources tree and select the *Open* command in the contextual menu which appears.



Follow the steps as shown in paragraph 2.8.1, until the new global variable appears as a new item in the pages tree.

2.8.3 IMPORTING THE PLC VARIABLES IN THE PAGELAB PROJECT

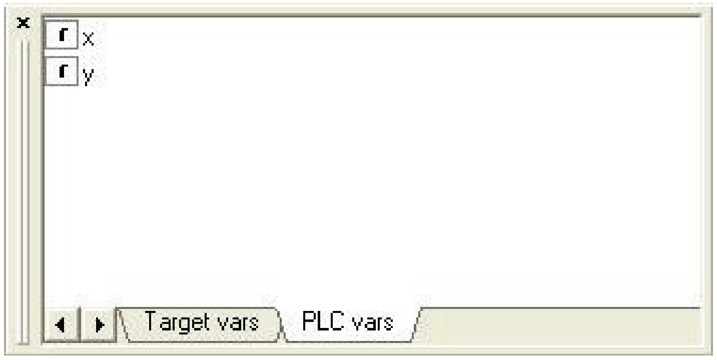
Usually an HMI project is not a stand-alone one, but is an interface for a PLC. More precisely, if the PLC project has been carried out with LogicLab, you can easily publish some variables to PageLab.

A variable of the LogicLab project can be exported to PageLab if it has been allocated on a datablock (it is not an automatic variable). If this pre-condition holds, when compiling the PLC, the program automatically creates an *.exp* file, which contains a list of the exported variables with their location in the datablocks, which the PageLab program can work out.

In order to import in PageLab the variables which have been exported from the PLC LogicLab project, you have to select the `Link PLC variables file...` from the `Project` menu.

A window opens and lets you select the file which contains the exported variables.

If you confirm to include the *.exp* file in the PageLab project, a new table called `PLC vars` appears in the libraries window. It contains the list of the exported variables.



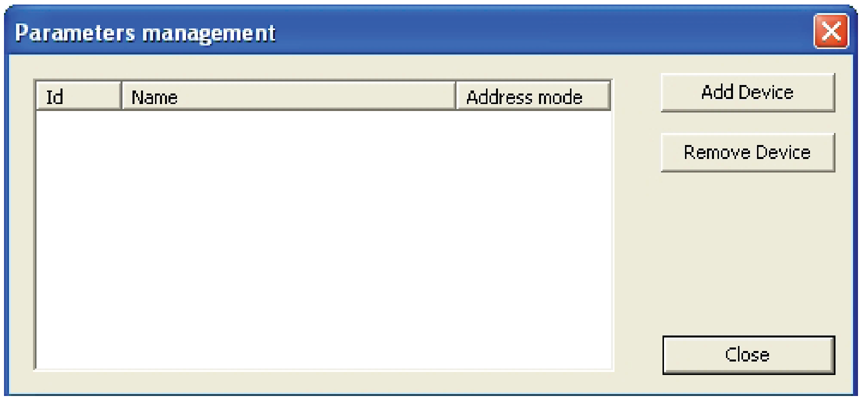
When you need to update the list of the exported variables, if the *.exp* file has not been moved to another directory, it is not necessary to repeat the above mentioned procedure. It is enough to launch the `Refresh PLC variables` command from the `Project` menu.

2.8.4 INSERTING FIELD PARAMETERS

Target system usually has internal variables and is connected on a fieldbus, so it needs to show some variables of the different devices which are connected on the net.

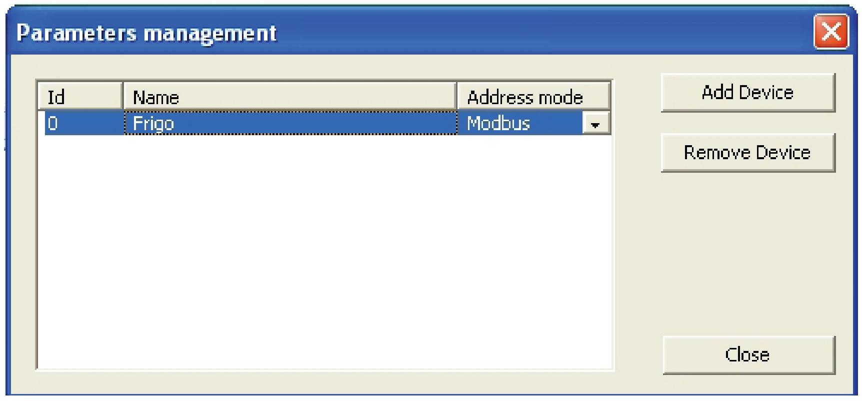
For this reason, PageLab lets you link a specific file which contains the variables definition on the bus. Click the apposite button in the toolbar.

The parameters management window appears.



Through the *Add Device* button you can add a new object linked to the target on the fieldbus.

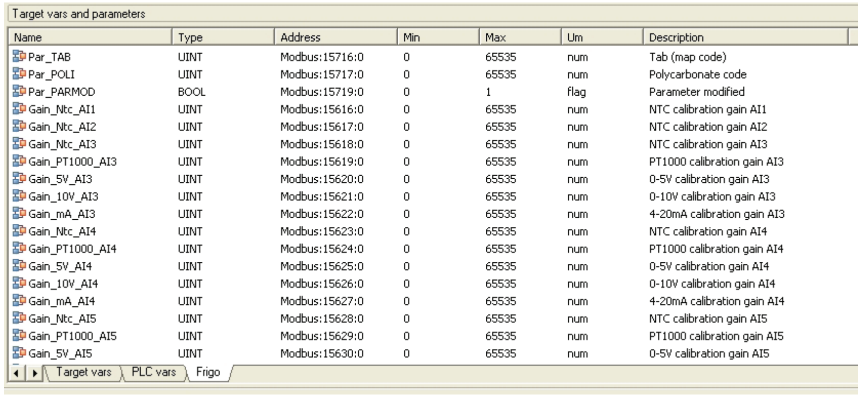
The selection window appears. Then you have to take from your PC a *.parx* file (see chapter 7). After inserting this file, the parameters management window will look like the image below.



Id	Name	Address mode
0	Frigo	Modbus

A device called *Frigo* has been inserted. In order to see the relevant parameters, click the *Close* button.

In the *Window target vars and parameters* you will see the device and its parameters.



Target vars and parameters

Name	Type	Address	Min	Max	Um	Description
Par_TAB	UINT	Modbus:15716:0	0	65535	num	Tab (map code)
Par_POLI	UINT	Modbus:15717:0	0	65535	num	Polycarbonate code
Par_PARMOD	BOOL	Modbus:15719:0	0	1	flag	Parameter modified
Gain_Ntc_AI1	UINT	Modbus:15616:0	0	65535	num	NTC calibration gain AI1
Gain_Ntc_AI2	UINT	Modbus:15617:0	0	65535	num	NTC calibration gain AI2
Gain_Ntc_AI3	UINT	Modbus:15618:0	0	65535	num	NTC calibration gain AI3
Gain_PT1000_AI3	UINT	Modbus:15619:0	0	65535	num	PT1000 calibration gain AI3
Gain_5V_AI3	UINT	Modbus:15620:0	0	65535	num	0-5V calibration gain AI3
Gain_10V_AI3	UINT	Modbus:15621:0	0	65535	num	0-10V calibration gain AI3
Gain_mA_AI3	UINT	Modbus:15622:0	0	65535	num	4-20mA calibration gain AI3
Gain_Ntc_AI4	UINT	Modbus:15623:0	0	65535	num	NTC calibration gain AI4
Gain_PT1000_AI4	UINT	Modbus:15624:0	0	65535	num	PT1000 calibration gain AI4
Gain_5V_AI4	UINT	Modbus:15625:0	0	65535	num	0-5V calibration gain AI4
Gain_10V_AI4	UINT	Modbus:15626:0	0	65535	num	0-10V calibration gain AI4
Gain_mA_AI4	UINT	Modbus:15627:0	0	65535	num	4-20mA calibration gain AI4
Gain_Ntc_AI5	UINT	Modbus:15628:0	0	65535	num	NTC calibration gain AI5
Gain_PT1000_AI5	UINT	Modbus:15629:0	0	65535	num	PT1000 calibration gain AI5
Gain_5V_AI5	UINT	Modbus:15630:0	0	65535	num	0-5V calibration gain AI5

When you need to update the list of parameters, if the *.parx* file has not been moved to another directory, it is not necessary to repeat the above mentioned procedure, but it is enough to press the button

2.9 INSERTING EDIT BOX

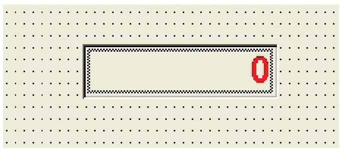
An edit box is a text frame which lets you display and eventually edit an associated variable or parameter.

2.9.1 INSERTING AN EDIT BOX IN THE PAGE

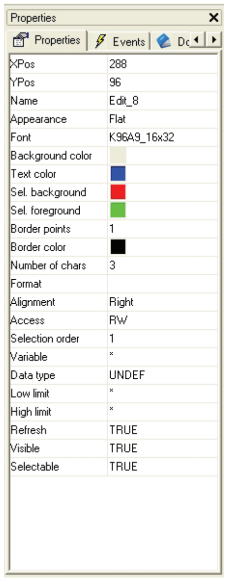
Insert an *Edit box* control in the page by pressing the corresponding button in the *Page toolbar*.

Move the mouse to the active area of the page. A cross + appears. The object will be inserted in the grid near to the mouse cursor.

Confirm the insertion point by left-clicking. A new text frame appears. It consists by default in a certain number of characters and its font is specified in the *Font* property of the page.



Edit this control's properties as you can see below.



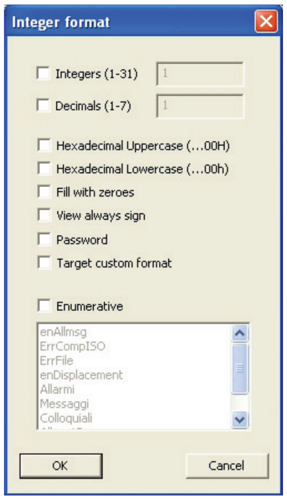
Properties

Properties | Events | Dc

XPos	288
YPos	96
Name	Edit_8
Appearance	Flat
Font	K96A9_16x32
Background color	⬜
Text color	🟦
Sel. background	🟥
Sel. foreground	🟩
Border points	1
Border color	⬛
Number of chars	3
Format	
Alignment	Right
Access	RW
Selection order	1
Variable	*
Data type	UNDEF
Low limit	*
High limit	*
Refresh	TRUE
Visible	TRUE
Selectable	TRUE

In the following list you can find all the changes which may be carried out:

- *Appearance*: you can make the edit box appearance "sunken" by assigning the *Sunken* property.
- *Font*: you can customize font by choosing, for example, a 16x32 font instead of the default 8x16 font.
- *Select background* and *Select Foreground*: respectively text and background colors when the edit box is selected.
- *Number of Chars*: maximum number of characters which can be displayed.
- *Access*: in order to set the read-only mode, replace *RW* (read-write) with *RO* (read-only).
- *Refresh*: in order to constantly update the contents of the edit box, select the *TRUE* option. Otherwise, the contents are refreshed just when drawing the page for the first time.
- *Format*: it represents the display format of the associated variable's value. The format value can be inserted only if a variable is just available. It opens a dialog window with these settings according to the type of variable (integer, real, string).



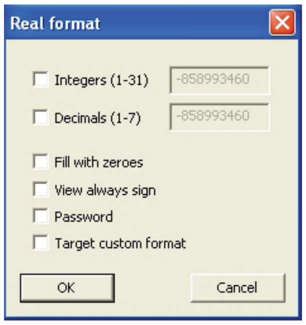
Integer format

- [ ] Integers (1-31) 1
- [ ] Decimals (1-7) 1
- [ ] Hexadecimal Uppercase (...00H)
- [ ] Hexadecimal Lowercase (...00h)
- [ ] Fill with zeroes
- [ ] View always sign
- [ ] Password
- [ ] Target custom format
- [ ] Enumerative

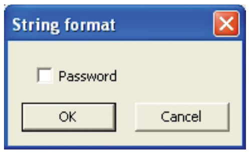
enAllmsg
ErrCompISO
ErrFile
enDisplacement
Allarmi
Messaggi
Colloquiali

OK Cancel

- *Integers*: number of digit before comma
- *Decimals*: number of digit after comma
- *Hexadecimal Uppercase*: the number is shown as 0...0H representation with uppercase H letter
- *Hexadecimal Lowercase*: the number is shown as 0...0h representation with lowercase h letter
- *Fill with zeros*: fill the entire editbox controls with 0 where there are not numbers
- *View always sign*: show the +/- symbol in editbox
- *Password*: show only * symbols
- *Target custom format*: the target can define custom format to show the data in a particular way. In that case there is a variable on the target with the value of the corresponding user mode.
- *Enumerative*: this representation allows to select a string value corresponding to numeric value defined in *Resources*, under *Enumeratives*.



- *Integers*: number of digit before comma
- *Decimals*: number of digit after comma
- *Fill with zeros*: fill the entire editbox controls with 0 whrere there are not numbers
- *View always sign*: show the +/- symbol in editbox
- *Password*: show only * symbols
- *Target custom format*: the target can define custom format to show the data in a particular way. In that case there is a variable on the target with the value of the corresponding user mode.



- *Password*: show only * symbols.

The *Target custom format* is a special feature which enables a particular custom format implemented on the target.

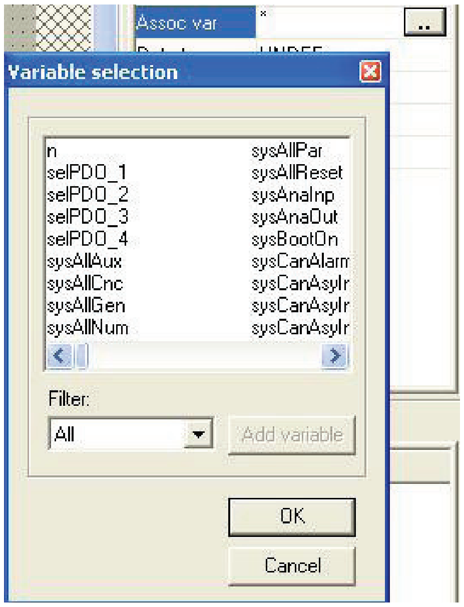
The format is specified according with language *printf* syntax (see 5.7.2).

2.9.2 EDIT BOX AND PAGELAB LOCAL VARIABLE ASSOCIATION

The edit box which you have just inserted lacks an essential element: the associated variable to take the values to display from. Let us assume that you want to link the edit box to a local variable (in order to get information on how to declare a local variable, see 2.8.1).

Select the edit box by clicking it once and select the `Variable` property.

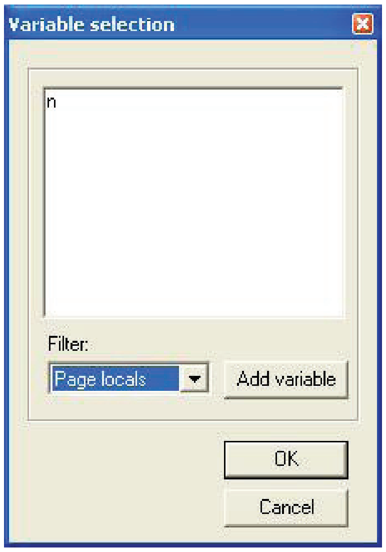
You can either type the name of the variable or click on the field and open the dialog window by clicking on the apposite button.



```
n             sysAllPar
selPDO_1      sysAllReset
selPDO_2      sysAnaInp
selPDO_3      sysAnaOut
selPDO_4      sysBootOn
sysAllAux     sysCanAlarm
sysAllCnc     sysCanAsyIr
sysAllGen     sysCanAsyIr
sysAllNum     sysCanAsyIr
```

Filter: All

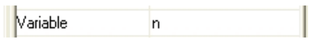
You can restrict the research just to the local variables of the `Init` page (consequently only the `n` variable) by using the `Filter` tool.



```
n
```

Filter: Page locals

Select the local variable. The `Variable` field in the table properties refreshes accordingly.



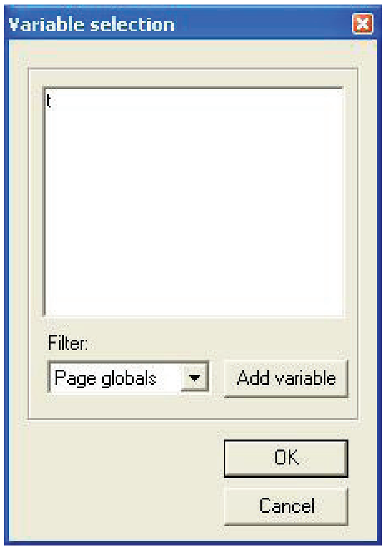
Variable	n

Then the `Edit box` control shows the `n` local variable's value constantly refreshed.

2.9.3 EDIT BOX AND PAGELAB GLOBAL VARIABLE ASSOCIATION

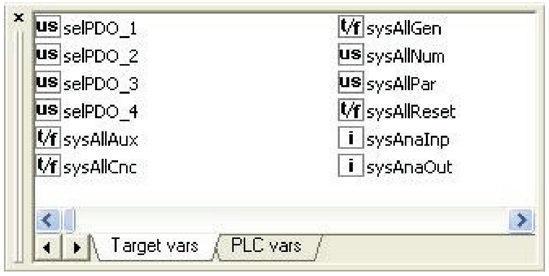
The principle to associate the `Edit box` control with a global variable is similar to the one to associate the `Edit box` control with a local variable. The difference consists in the variable declaration (in order to get information on how to declare a global variable, see § 2.8.2).

You can associate the Edit box with the global variable through the dialog window which was introduced in the preceding paragraph, but in this case it is necessary to use a different filter in the `Filter` field.

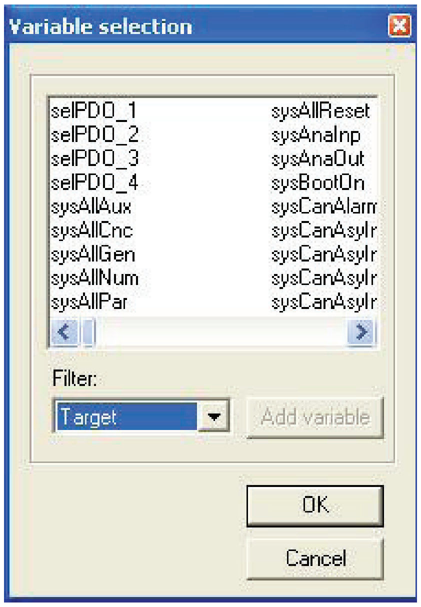


2.9.4 LINKING AN EDIT BOX WITH A TARGET (OR SYSTEM) VARIABLE

The target system executing PLC and HMI often publishes some variables which allow the interaction between user interface and system. In PageLab, such variables are called target variables. You can view them in the `Target vars` table of the `Target Vars and Parameters` window.

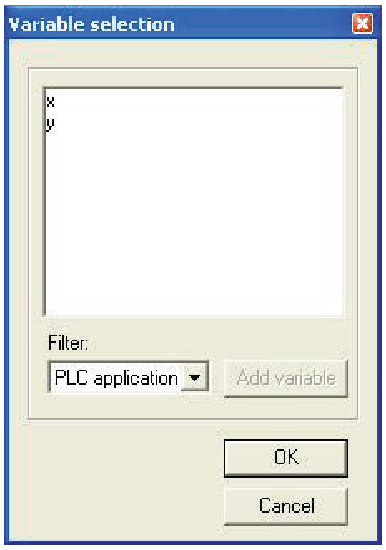


You can associate an Edit box with a target variable through the dialog window which opens from the `Variable` field, but in this case it is necessary to use a different filter in the `Filter` field.



2.9.5 LINKING AN EDIT BOX WITH A PLC LOGICLAB VARIABLE

You can associate an Edit box with a PLC LogicLab variable through the dialog window which opens from the `Assoc var` field (see 2.9.2), but in this case it is necessary to use a different filter in the `Filter` field.



2.9.6 LINKING AN EDIT BOX TO A PARAMETER

You can associate an Edit box with a parameter through the dialog window which opens from the `Variable` field (see 2.9.2), but in this case it is necessary to use a different filter in the `Filter` field.