

SAIA®PCD
PLC based terminals

SAIA HMI-Editor
Version 1.4.100
User manual

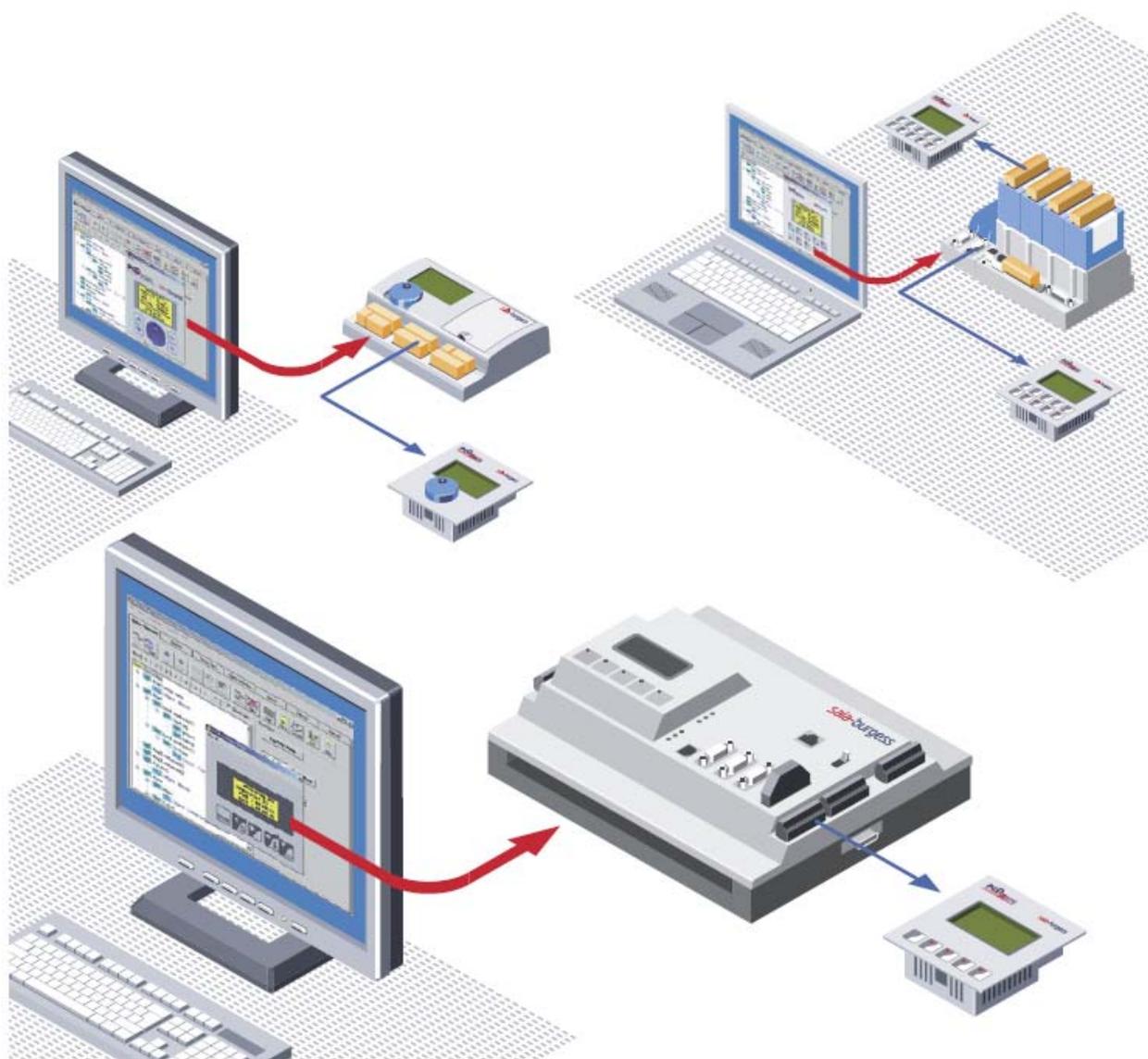


Table of Contents

1	GENERAL	5
1.1	WELCOME.....	5
1.2	KEY WORDS AND BASIC NOTIONS.....	7
1.3	GENERAL CONCEPT	10
2	COMMANDS	11
2.1	COMMANDS	11
2.2	IMPORT	12
2.3	LANGUAGE MANAGER.....	15
2.4	PLAY FEATURE.....	19
3	MENU STRUCTURE	20
3.1	MENU STRUCTURE TAB	20
3.2	ROOT MENU	22
3.3	ROOT MENU EDITION	24
3.4	CALL TYPE (MENU STRUCTURE).....	26
3.5	FUNCTION KEYS (MENU STRUCTURE).....	28
3.6	ACCESS & RESTRICTIONS (MENU STRUCTURE).....	29
3.7	OBJECT PREVIEW (MENU STRUCTURE).....	30
3.8	OBJECT VIEW (MENU STRUCTURE).....	31
4	OBJECTS.....	32
4.1	OBJECTS TAB	32
4.2	EDIT OBJECTS	33
4.3	ADVANCED OBJECTS	35
5	STATUS BARS TAB.....	37
6	ATTRIBUTES	38
6.1	TEXT ATTRIBUTE	39
6.2	TYPE ATTRIBUTE.....	40
6.3	FORMAT ATTRIBUTE	41
6.4	STATUS ATTRIBUTE.....	43
6.5	ACCESS ATTRIBUTE	44
6.6	UNIT ATTRIBUTE.....	46
6.7	MIN/MAX ATTRIBUTE.....	47
6.8	DEFAULT RESOURCES.....	48
6.9	RESOURCES	49
6.10	SYSTEM TEXTS.....	52
6.11	ICONS.....	53
7	ALARMS.....	56
7.1	ALARMS TAB	56
7.2	ALARM SETTINGS.....	58
8	SETTINGS.....	61
8.1	SETTINGS TAB.....	61
8.2	TERMINAL SETTINGS	62
8.3	TOPOLOGY	63
8.4	SERIAL LINE SETTINGS	65
8.5	TERMINALS LIST	66
8.6	SETTINGS - OPTIONS	67
8.7	PASSWORD SETTINGS	68

9	INTERNAL.....	70
9.1	INTERNAL TAB	70
10	HOW TO ?.....	71
10.1	BUILD THE MENU STRUCTURE.....	71
10.2	CREATE THE PCD PROGRAM.....	72
10.3	IMPORT STANDARD OBJECTS	73
10.4	USE STANDARD OBJECTS.....	75
11	INSTALLATION AND OPTIONS.....	77
11.1	ADD-ON TOOL INSTALLATION IN PG5	77
11.2	EDITOR OPTIONS	78

1 General

1.1 Welcome

SAIA HMI Editor. Version 1.4

Welcome to the SAIA HMI Editor. This tool allows you to create an HMI application for SAIA text terminals. With the HMI Editor you will create an HMI menu tree, define password levels, allow display and editing of variables, access internal Fbox parameters and create an alarm list.

Read first the topic: [Keywords and basic notions](#) to become familiar with important expressions used in the HMI Editor.

The understanding of the [General Concept](#) of the HMI Editor is essential to work efficiently with this tool.

You will find a description of each main tab in the following topics:

- [Menu Structure](#)
- [Objects](#)
- [Object attributes](#)
- [Alarms](#)
- [Settings](#)
- [Internal](#)

To start working with the HMI Editor, consult the following 'How to...' topics:

- [Build the menu structure](#)
- [Create the PCD program](#)
- [Use standard objects](#)

Restrictions, Requirements, Recommendations

The present version of the HMI-Editor requires PG5 version 1.4.xxx.

When several terminals are used, all terminals are of the same type.

Only one terminal at a time can be used. When a user is logged on a terminal, the other are put in a busy state.

The Language Manager supports up to 5 languages.

The menu tree can have up to 8 levels with each up to 20 positions (20 * 20 * 20 * 20 * 20 * 20 * 20 * 20 positions). Take care that the performance of the editor becomes limited with about 500 menu positions.

Some features are not supported with the bus topology :

- Bargraph display
- LED control
- Beeper control
- Icon and Status bar
- Trend advanced objects
- No Flags are controlled by F-Key (menu calls only)

Restriction with handshake options: It cannot be ensured that the display work correctly when **no** handshake is used. In case of problem the only solution is to reduce the baudrate. The same restriction as for the bus topology (see above) is valid with XON/XOFF handshakes.

In all case when using PCD7.D23x terminals, read in first the user's guide (in the packing), the TI 26/430 or the manual (26/795).

The alarm list can hold up to 500 alarms. Take care that the a long list of alarms can drastically slow down the user program.

Some conditions apply to the alarm buffer, especially with a size higher than 50 entries. See the topic [Alarm settings](#)

Flags and Registers can be used as variables. Other media (Inputs / Outputs / Counters and Timers) can be used with recent firmware only. In this case the special option 'All media' must be set in the Settings tab. If Hevac or Modem objects are used, the special option 'All media' must also be set in the Hevac Init Fbox and the Modem Driver.

Flags and Register can be used as variables. Other media (Inputs / Outputs / Counters and Timers) can be used with recent firmware only (see chapter 8.6).

The terminal type D230 applies to the external PCD7.D230 display as well as to the internal display of the PCS1 systems.

Advantages and new features compared to version 1.3 .

No new features are currently added in the version 1.4.

1.2 Key words and basic notions

Root menu

The root menu is the first menu displayed on the screen when the PCD is powered up, restarted or the terminal is not in use. Up to 3 root menus are displayed depending on the used topology.

Single terminal: Standby menu

Multipoint terminals: Standby menu -> Busy

Bus terminals: Standby -> Busy -> Login menu

When the user is logged on a terminal, the busy status is displayed on other terminals.

After a timeout, if no key is pressed, the Standby menu is displayed again.

Menu call type

Each menu position represents an entry point for the next dialogue level. The type of the next level (reached by pressing the enter key on the terminal) is defined by the "Call type".

Password

The password is entered from the Login menu. A list of 10 passwords is defined with the Editor. Each password is associated to a level. 0 is the lowest level (less rights) while 9 is the highest level (all rights). On each menu position you can define a minimum password level and a restriction if the password level is to low.

Objects

An object is a collection of elements. Objects are placed at the lower level of the menu tree. In an HMI project, you can prepare a set of objects that you may use several times in the menu tree. Therefore you must first define objects and then reference them in the menu tree.

Standard Objects

Standard Objects are predefined objects for Fupla Fboxes. You don't need to build the standard objects, you can import them from a standard file (*.LMI file) delivered with the Fbox library. In the Tree view, you will directly call them by specifying their Fbox names.

Advanced Objects

[Advanced Objects](#) is a feature that allows to import special predefined object in the HMI. An advanced object is not limited to strict structure of 1 to 20 element lines. The HMI library includes 2 default advanced objects supporting trend display for the D23x. Further advanced object can be developed on requests.

Status bar

Status bars can be defined for Root, Menus, Objects and Alarm buffer. Status bars can use static or dynamic icons. The defined bars will be placed at the 2 bottom lines of the display. Dynamic icons can be switched by mean of a flag or selected over a register value.

Elements

An element is represented by one line in an object. Usually it allows you the display (optionally the modification) of a variable on the terminal. Elements can only be placed in objects, not in a menu. To build an element you need some attributes like text, value format, access and limits). The resources (registers and flags) associated to the elements are not defined in objects. You will define various resources for each instance of an object in the menu tree.

Objects Attributes

The attributes are the basic 'material' you will use to build element lines. Therefore you have to define a set of attributes before building your elements and objects. Some default attributes are already defined when starting a new HMI application. You will have to add more attributes for your specific application. For instance Texts, Status and Units. Other attributes are bound to the HMI routines in the PCD and therefore cannot be freely extended by the user. For instance access mechanisms and variable types.

Menu texts

Menu texts are directly edited on the menu tree for each menu position.

Text lists and texts

You will enter all texts that you need to define elements with, in the text lists. Use the text lists organization to group texts according to your favorite categories (by language, theme or application).

Status

Statuses are strings that will be displayed in place of a value on the terminal. An element using status will need a list of statuses corresponding to the valid value range.

Min/Max Limits

For modifiable values, you can define value limits. Limits are not defined individually to each element. The limits set can be defined once and used on several elements.

Resource lists and Resources

Resources are not associated to each object definition. Each object instance will have its own resources. Therefore you must associate resource to objects that are called from the menu tree. You may use physical addresses or symbols from the global symbol list.

Icons

Icons are small pictures of 3 characters wide and 2 lines high. Icons can be created by the user with the Icon Editor and used in the HMI file. A free number of icon libraries can be created or received from SAIA or third party. Up to 100 icons can be imported in an HMI file and then loaded in the D23x terminals. Icons can be used in menu (static icons) and in Status bar (dynamic icons).

Value type

Type attribute defines the value type in the PCD. For example binary or integer. The type can also define a value conversion. For instance, date stored as MM.DD but displayed as DD.MM (DD=day, MM=month).

Internal resources

Some internal resources may optionally be accessed by the user. For example, the password list. LED flags and F-Key flags are also implemented as internal resources. These variables can be accessed by System symbols in any other module (IL, S-Fup or the HMI itself). The system resources are displayed in the Internal tab and are also listed by the Symbol editor after a first compilation. It is the responsibility of the user to access correctly to these resources. For example, F-Key flags must be read and reset when the F-Key command is executed.

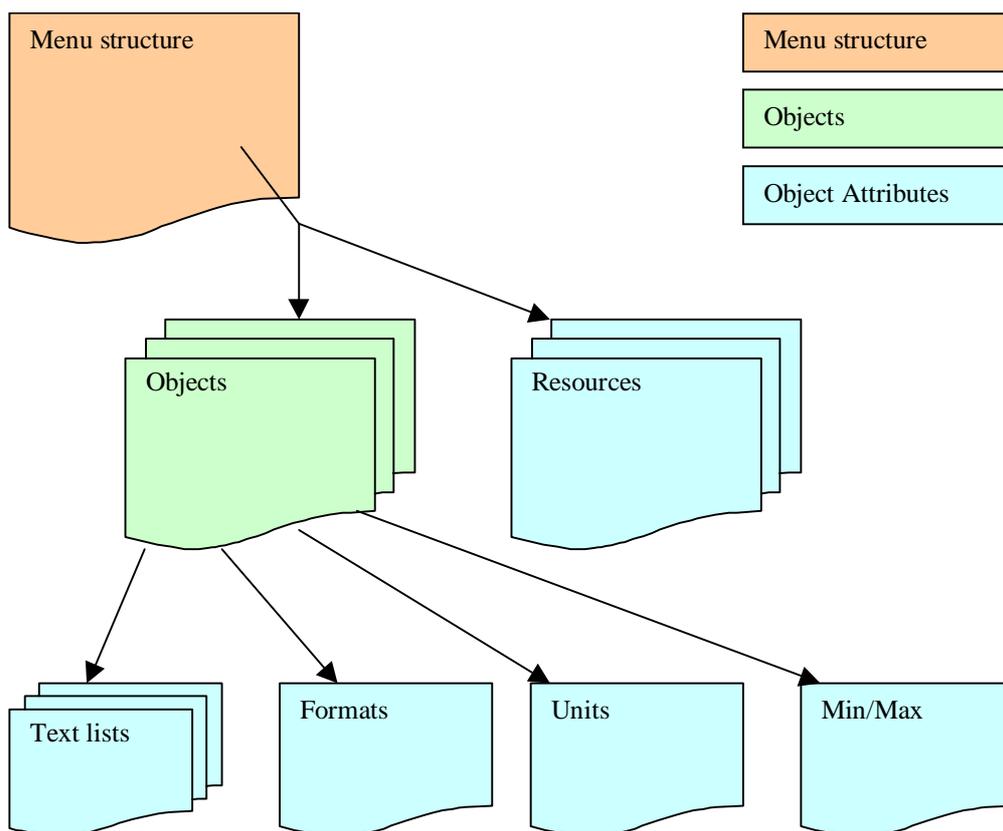
1.3 General Concept

The HMI application is built by nested references starting from the Root menu. The root menu points to further menus pointing to sub-menus and so on. This tree builds the Menu structure.

At the lowest level of this tree the last menu position points to one object representing in fact an object instance. In the menu tree view the object instance appears in a compact form. On the right pane the object instance is displayed with the resource references. Therefore, variables will only appear at the lowest level of the structure, actually in the objects.

An object at its turn, is made of 1 to 20 lines pointing to various attributes. Attributes are collected in attribute lists and stored once in the project.

Each menu position with all its properties and the object instance with its resources are stored in the menu structure for the project. The object structures are stored only once in the project file, even if used several times.



2 Commands

2.1 Commands

File Menu

New	Ctrl+N	Open a new HMI file. Note that only one HMI file can be linked with a project.
Open..	Ctrl+O	Select and open a HMI file.
Save	Ctrl+S	Save the current HMI file.
Save as		Save the file under a new name.
Restore Last saved.		Restore the content to the latest saved version of the file.
Play	F11	Start the play feature
Compile	Ctrl+K	Compile the current HMI file.
Import		Import from an existing HMI file.
Import standard Objects		See Import standard objects
Import advanced Objects		See Import standard objects
Exit		Exit the HMI Editor.

View Menu

Menu structure	View of the Menu Structure tab
Objects	View of the Objects tab
Status Bars	View of the Status Bars tab
Object Attributes	View of the Object Attributes tab
Alarms	View of the Alarms tab
Settings	View of the Settings tab
Internal	View of the Internal tab

Languages

Default, others...	List of the available languages and selection of the current language.
Remove...	Remove languages
Add New...	Add new languages
Review	Open the tool to review the languages.

Options Menu

Editor Options	Open the dialog to set the editor options .
----------------	---

Help Menu

Help Topics	Call the HMI Help.
About	Open the about dialog

2.2 Import

The Import command allows you to import part of an existing HMI file into the current file. You can choose to clear the current file or not before the importation.

Importation without clearing the current file is a critical operation and you should carefully check the result after the operation. Typically, the importation is made at the beginning of the creation of a new HMI application. If the current file already contains an important structure, it is recommended to save the current file and to work on a new copy of it.

The import command will first open a dialog to brows and select the HMI file to import. Then a dialogue is displayed to allow you the selection of the parts you want to import. Your selections are saved are restore each time the Import called. The 'Restore' button allows you to recall the last used options.

For parts that cannot exist several times in the file, the current data will always be overwritten. It is the case for :

- Root Menu
- Settings
- Internal Resources

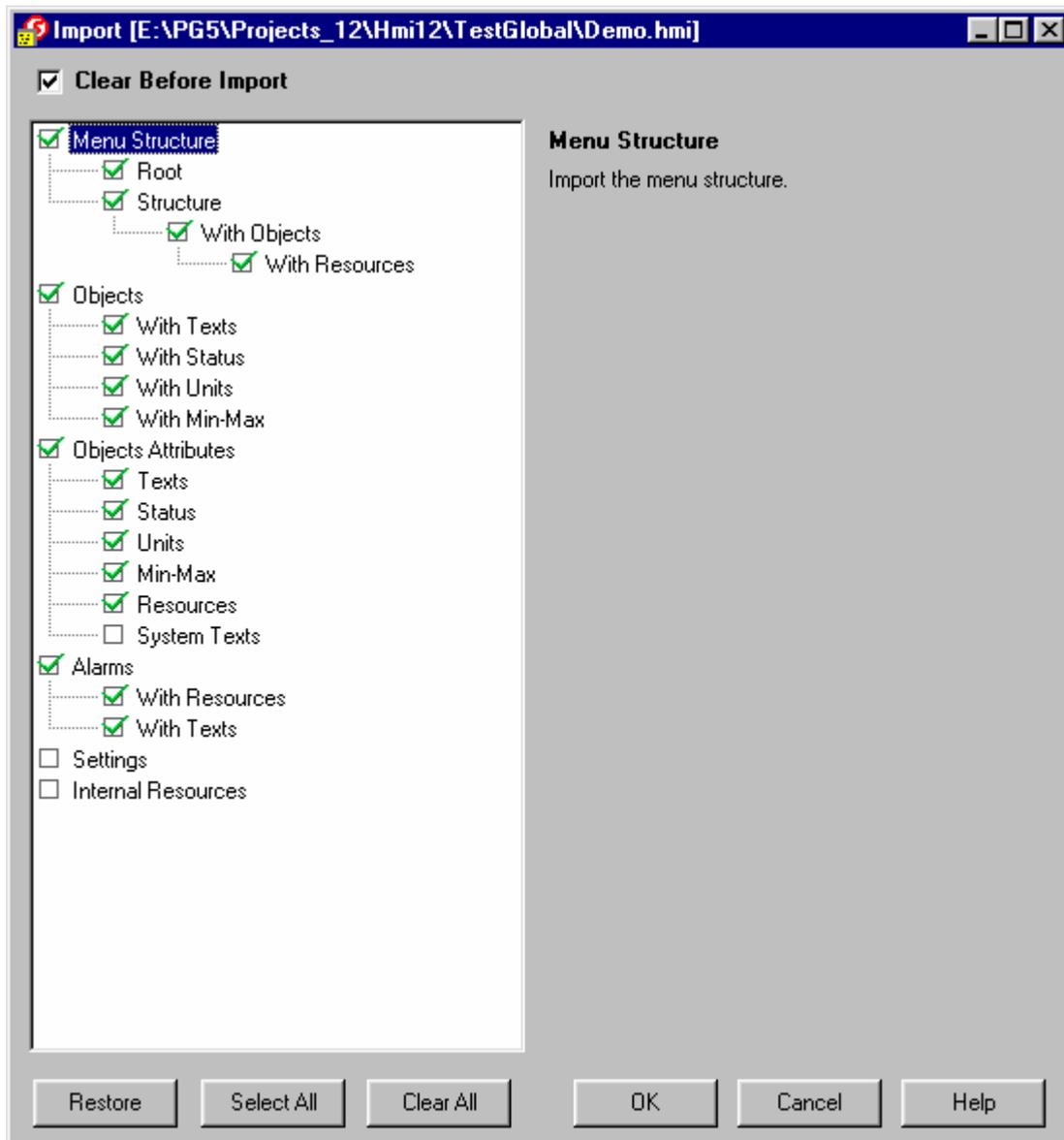
The importation is subject to two important restrictions:

File version

Importation can only be made between files of the same version. Therefore, you may be requested to convert the file to import before to be able to use it for importation. To convert a HMI file to the new version, simply go to the concerned project and CPU, open the HMI file, and confirm the question to convert, save and close the new file.

Standard objects

Standard objects cannot be duplicated in a HMI file. Therefore, it is not allowed to import objects if both, the source and the current files, include Standard objects. You should import from the HMI file before to load the standard objects into the current file.



For each part of the HMI file a short description is displayed on the right side when you click on it. This help you to understand what is concerned by each option.

As you check options, you will notice that other options are automatically selected because they are connected together and also requested. When you uncheck options, in the opposite, no other are unchecked. It is your task to uncheck all parts you don't want to be imported.

Languages

The HMI file can contain texts in several languages. Depending on the number of languages in each file the following situations can appear:

Source file	Current file	Comment
One language	One language	The text are imported without option.
One language	Several languages	The same texts are imported in all

Several languages	One language	languages. You are prompted to select which language, you want to import.
Several languages	Several languages	You are prompted to select which languages, you want to import. Only the languages with matching names can be imported. If required, rename the languages to match in both files.

Take care that if you import several texts successively, you will get new text lists and new text on each import. You cannot import new languages in existing texts if you do not import it in a single operation .Menu Structure.

2.3 Language Manager

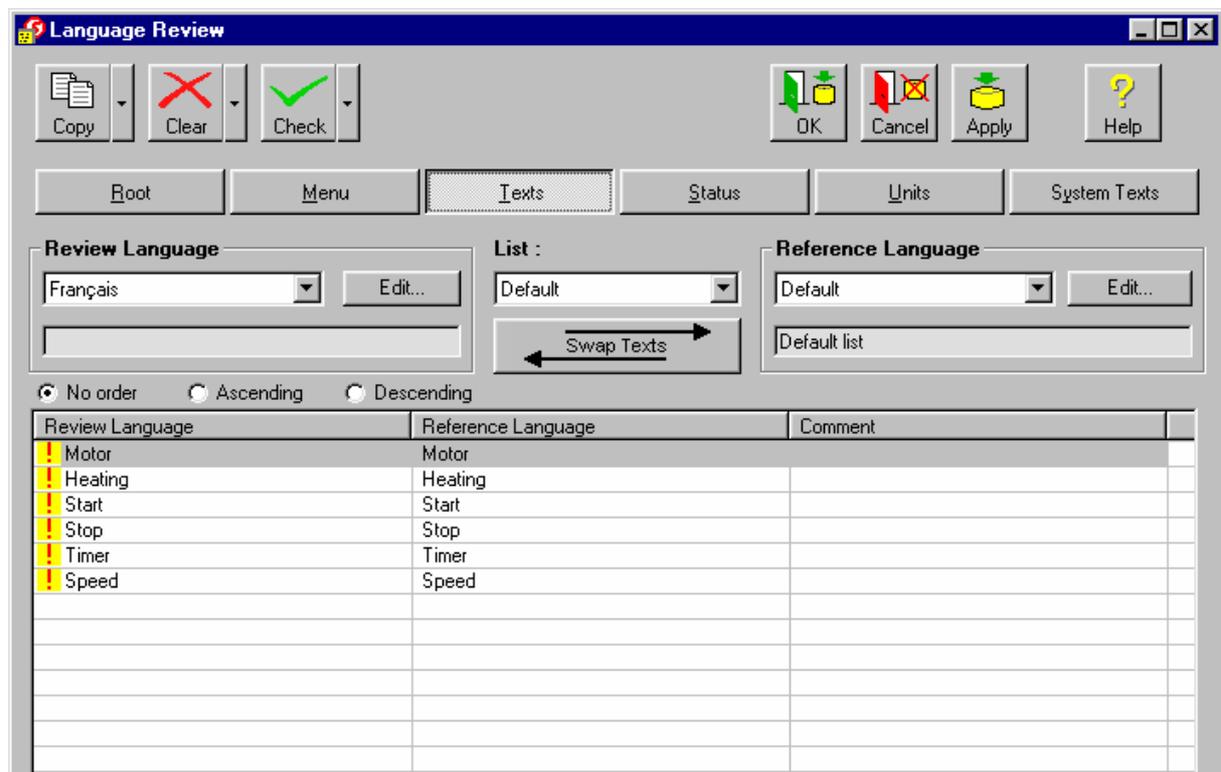
The Language manager allows you to enter all texts in up to 5 languages in the HMI file. The selection of the displayed language on the terminal is made before compilation. In this way, you can create a new application in another language by just recompiling the same HMI file.

The dialogue structure is always the same for any language. Only the last selected language texts will be stored in the PCD memory.

When you create a new HMI file you have only one language (Called Default language) and you work with a single language without any special care (as with former version).

To work with several languages, you have to know the following points :

- You must first create a new language using the command ' Add New...' in the 'Languages' menu.
- You always work, as before, on a current language. By default you new texts will be added to all languages. You will have to translate your text later.
- You can select the current language in the menu 'Languages'. But this is not the easiest way to translate you texts.
- You can use the special tool 'Language Review' to review your texts and translate in several languages. Use the command 'Review' in the 'Languages' menu.



In the Language Review tool you can review and translate your text in a comfortable way. All text from the HMI file are available by the buttons Root, Menu, etc...and the list selection. You always work on one language on the left side, called the 'Review Language' while another language, called 'Reference Language' is shown on the right side. You can change the Review Language and the Reference Language at any time.

When you have create new texts, they have been automatically copied in all languages as well as when you have created a new language, you have maybe select the option to automatically copy all texts to the new language. These texts have not yet been translated and are marked with a red warning sign . It show you which texts must still be translated or reviewed. Once reviewed, the mark automatically turns to a green OK sign .

Elements of the Review Language tool



Copy the texts from the Reference Language to the Review Language. By using the pull down menu, you can choose which part of the texts you want to copy.



Clears the texts from the Review Language. By using the pull down menu, you can choose which part of the texts you want to clear.



Checks (change the warning sign to the check sign) the texts from the Review Language. By using the pull down menu, you can choose which part of the texts you want to check.



Accept all the changes you have made, save them to the internal database (not yet to the disk) and exit the Language Manager.



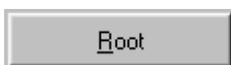
Ignore all the changes you have made and exit the Language Manager.



Save all the changes you have made to the internal database (not yet to the disk) but don't exit the Language Manager.



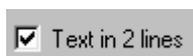
Call the Help system about the Language Manager.



Buttons to select the texts to review. Similar for Root, Menu, Texts, Status, Units and System Texts.



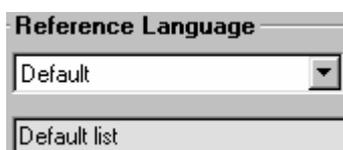
Selection of the text list where applicable.



This checkbox is shown when you edit the menu texts. When the 2 lines menu are used, check this box to see the 2 lines of the menu texts.



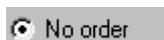
Selects the language to review.



Selects the reference language. The line below the selection show the comment associated to the language.



Button to edit the language name and the comment associated to it.



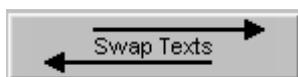
The texts will not be sorted in the list.



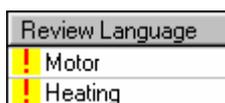
The texts will be sorted in ascending order in the list.



The texts will be sorted in descending order in the list.



Swap (exchange) the Review Language with the Reference Language. Note that you are normally working on the Review Language. If you see a need to modify texts of the Reference Language, you can quickly exchange them with this button.



List of the selected text list of the Review Language. The language you are normally working on. You can edit this text in many ways. When selected, a new click open the in-place edit field. A double-click open a new window where you can also edit the Reference text and the comment. The Alt+Enter keys open the in-place edit field. The Ctrl+Enter keys open the edit window.



Warning sign for texts that are not yet been reviewed. This sign is created when a text is automatically copied. This append when you enter new texts in the Object Attributes tab and also when you create a new language.



OK sign for texts that have been reviewed. This sign is automatically displayed after a text have been edited in the Review Language tool.

Reference Language
Motor
Heating

List of the selected text list of the Review Language. You are normally not working on this text but you can also edit it in the edit windows. Double-click on the text or press Ctrl+Enter.

Comment

Comments of the selected text list of the Review Language. The comment is for the developer and it is the same for all languages (don't translate it). You can edit it in the edit windows. Double-click on the text or press Ctrl+Enter.

2.4 Play feature

The play feature allows you to preview your work with the Editor without having to create and run the PCD program. You will get a better feeling on what will be displayed on the screen. Only the menu navigation is supported (no edition and no value refresh). The passwords restrictions are also not applied. The terminal picture used for the play feature depends on the 'Play as' selection in the [Settings tab](#).

The Play feature is started using the Play button in the [Menu Structure](#) tab.

Example:



3 Menu Structure

3.1 Menu Structure tab

In the menu structure you can build the menu tree, edit menu texts, call objects and define password levels.

In the left part you have an overview of the edited menu structure. Starting from the [Root menu](#), you can edit the menu tree by adding new menu lines and child lines.

To get a better overview of a complex structure, you may restrict the displayed levels using the tabs 1 to 7. For a concentration on menu position only, uncheck the Object view. You may individually show or hide any menu by toggling the + / - sign of each menu position.

When selecting a menu line, you will have the [Call type \(Menu Structure\)](#), the [Function keys](#) and the [Password](#) frames. When an object is selected you will have an [Object frame](#) and alternatively, the [object preview](#)

When the root menu is selected in the menu tree, you have an overview of the various displays for the root menu. Depending on the selected topology, up to tree root menus are available: Standby, Busy and Login. These menus can be modified in the [Root Menu Edition](#) window by clicking the 'Edit Root Menu' button.

Tool bar :



Add a new position in the current menu.



Add a child menu at the current position.



Moves the selected menu position Up.



Moves the selected menu position Down.



Copy the current position with submenu and object to the clipboard.



Cut the current position with submenu and object and copy to the clipboard.



Paste the content of the clipboard at the selected position.



Delete the current position with all submenus and objects.



Clear the whole menu structure and rebuild an empty content with the root menu only.



Start the Play feature.



Save the current HMI file.



Compile the current HMI file.



Exit the HMI Editor.



Call the HMI Help.

3.2 Root menu

The root menu is the first menu displayed on the screen when the PCD is powered up, restarted or when the terminal is not used. Depending on the selected topology, up to three root menus are available : **Standby**, **Busy** and **Login**.

The **Standby menu** is always used when no dialogue is running.

In **Single** topology, the user can directly enter the password and press Enter or an F-Key.

In **Multipoint** and in **Bus** topology, the user must first press a key on a terminal to get the Login menu. At the same time, the other terminals will get the Busy menu. Therefore, the password is not entered in the Standby menu but only in the **Login menu**.

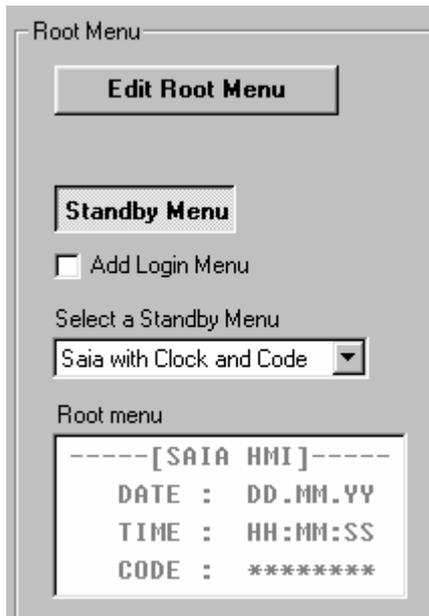
With the option '**Add Login Menu**', you can force to use 2 screens, the Standby and the Login even with a **Single** terminal.

The busy menu is used with the bus and the multipoint topologies when a user is logged in on another terminal.

Some predefined root menus are always available. The Custom option defines a free configurable root menu. The 'Custom' root menu can be edited line per line and some variable fields can be displayed.

If you need the full flexibility to build your own text, use the option 'Free Standby Text'. Enter a text address in the field for PCD Text and create your text in the Global symbol list.

Doing, so you can use all text commands supported by the PCD firmware (see PCD instruction manual).

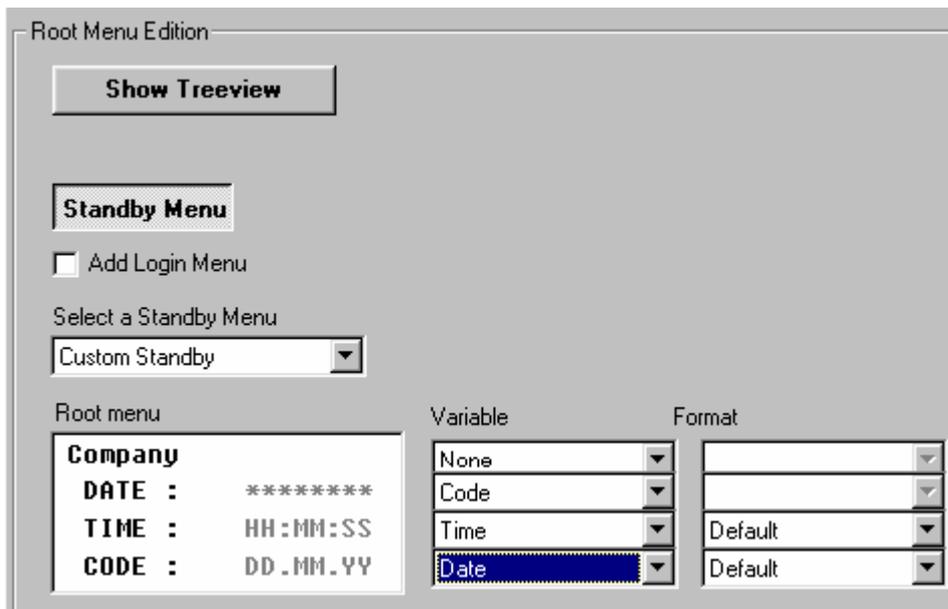


The root menus can be modified by clicking the ['Edit Root Menu'](#) button.

3.3 Root Menu Edition

The Root Menu Edition display is reached by clicking the Edit Root Menu button when the Root menu is selected in the tree view.

The display of the [menu structure](#) is called back with the 'Show Treeview' button.



If you select a variable from a line, you can edit the small text preceding the variable field. It is equal to half of the screen size. The rest of the line is used by the variable. If no variable is selected, you can edit the complete line.

The available variables are:

- Time PCD Clock time in format HH:MM:SS (hour, minutes and second).
- Date PCD Clock date in format DD.MM.YY (day, month and year).
- Code The password edition is displayed. The format cannot be changed here. It is selected in the Password Settings (Settings tab).
- Variable 1 to 4 An array of 4 PCD registers can be specified in the Advanced tab for this display option. The option selects the variable within the array of 4 registers. The format can also be selected.
- Variable 4 ASCII The 4th variable can be displayed in ASCII format with this option. The user program must ensure that the register always contains the ASCII codes of 4 displayable characters.
- Variable 3+4 This combine the variable 3 and 4 in ASCII format. It allows

- | | |
|-------|---|
| ASCII | to display a variable text with 8 characters. |
|-------|---|
- | | |
|--------------------|---|
| • Alarm indication | Shows 'ALA' if one or more alarms are currently active.
Else shows 'OK'. |
|--------------------|---|
- | | |
|------------------|---|
| • NAK indication | Shows 'NAK' if one or more alarms are not yet acknowledged.
Else shows 'OK'. |
|------------------|---|
- | | |
|-----------------------|---|
| • Variable 3
ASCII | The 3 rd variable can be displayed in ASCII format with this option. The user program must ensure that the register always contains the ASCII codes of 4 displayable characters. |
|-----------------------|---|

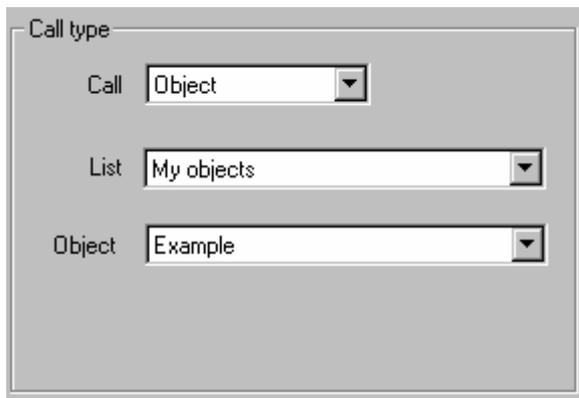
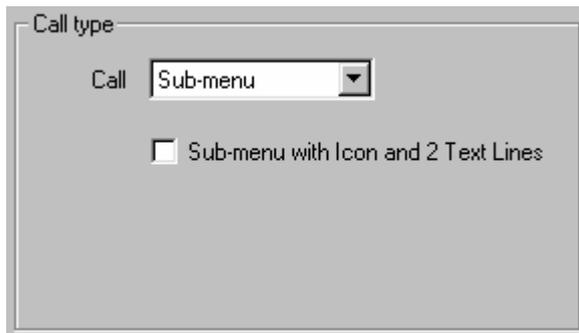
With the 'Free text' option, it is possible to use your own PCD text to be displayed. The PCD text will be cyclically sent to the terminal with a 'Cursor home' command. The free PCD text should not include terminal commands able to alter any terminal settings (character set, cursor settings, display mode, and so on).

3.4 Call type (Menu Structure)

In the Call type frame you can change the call type and select the called object.

A menu can call another menu over up to 8 nested levels. A menu can also call an object and with this define the lowest level of the menu tree. The root menu has default "Call type" and it always calls a menu.

For the call type 'Object', the object to be called must also be defined.



Call:

Sub-Menu:

This menu position will call another menu.

Object:

This menu position calls an object. The lists of objects in the project will appear.

Alarm buffer

This menu position will call the alarm buffer. The alarm buffer option must be checked in the Alarms tab. No other parameters are needed.

First menu

Returns immediately back to the first menu level.

Root menu

Returns immediately back to the root menu and reset the entered password.

For Sub-menu call, the option 'Sub-menu with icon and 2 text lines' appears if this function is supported by the selected terminal (e.g. D23x). By checking the option, the called menu will

3.5 Function keys (Menu Structure)



In the Function keys frame you can select an F-key call for the selected menu, the object or for the alarm buffer.

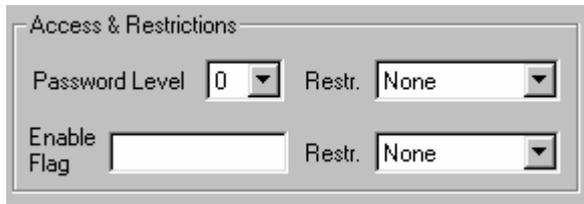
The terminal option is useful in the case of the bus or the multipoint [topology t](#). It defines for which terminal the F-Key call is restricted. Using the terminal option it is possible to assign the same F-Key to different menu or object for each terminal.

For instance, assuming that a terminal is installed in each room of a building floor, a specific object is created for each floor. The F1 key is assigned, for each terminal to the object corresponding to its room. In this way, on each terminal, the user can call the specific object using the same F-key, F1.

If the same F-key (and the same terminal option) is assigned to several menus or objects, each F-key selection will jump step by step to the next corresponding position and, after the last, comes back to the first position.

In the [password](#) menu, it is also possible to enter the password and to press directly an F-key in place of the Enter key. The password is checked and the F-key call is executed if a valid password has been entered.

3.6 Access & Restrictions (Menu Structure)



With the password level option, you can select the minimum password level required to enter the nested level of the selected menu (may be a submenu or an object). The Restriction option defines the applied restriction when the password level is to low.

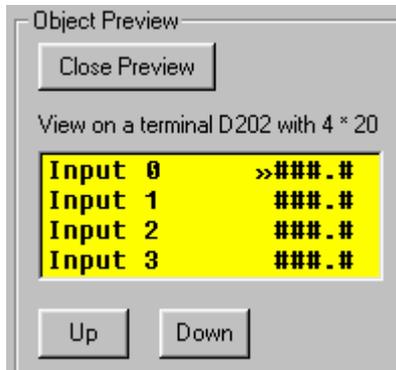
With the Enable flag your apply restrictions in runtime by setting or resetting the flag. The Restriction option defines the applied restriction when the flag is reset.

None:	No restriction.
Read only:	Should only be used for an object call. The called object is displayed but no edition will be allowed.
Avoid access:	Entering the menu or the object will be avoided.
Hide menu:	The menu position is not visible to the user.

The 10 passwords are defined in the [Settings](#) tab.

3.7 Object preview (Menu Structure)

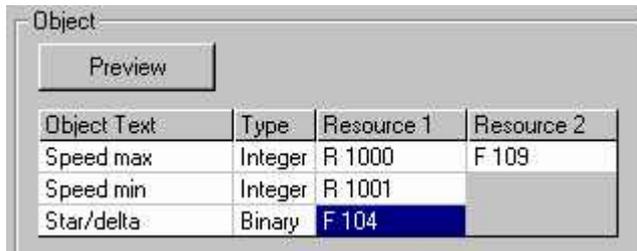
The object preview button give a quick preview of the object on the selected terminal.



The object is displayed on the selected terminal. Use the correct terminal type to have the best preview of the result of your work. Note that not all terminals can be previewed. Therefore you must use the 'Play as' option in the [Settings](#) tab.

The [Play](#) feature also allows you to navigate through the menu tree and will give you a better feeling on the runtime behavior of the terminal.

3.8 Object view (Menu Structure)



The screenshot shows a dialog box titled 'Object' with a 'Preview' button and a table. The table has four columns: 'Object Text', 'Type', 'Resource 1', and 'Resource 2'. The 'Star/delta' row is selected, showing 'Binary' as the type and 'F 104' as Resource 1.

Object Text	Type	Resource 1	Resource 2
Speed max	Integer	R 1000	F 109
Speed min	Integer	R 1001	
Star/delta	Binary	F 104	

The object view allows you to see and select the resources associated to the selected object. Resources are selected from the resource list. Go to the [Object Attributes tab](#) to create your resource list.

In this view, the text and the variable type are shown to help you selecting the right resources. The type is only applicable to Resource 1. Resource 2 is always binary (flag).

The **Resource 1** is the main resource used to display the variable on the screen. For a binary type select a flag. For a numeric type, select a register.

The **Resource 2** is only needed for specific Access type using an auxiliary flag like Override, Action or Update. Resource 2 is only activated if required for the access mechanism of the particular element.

It is the task of the PCD user program to execute the specific interaction between the variable and the auxiliary flag. See also the description of each [Access type](#).

4 Objects

4.1 Objects tab

The Object tab allows you to create object lists, define the free objects and import standard objects. The left pane shows the existing lists and objects in the file. The right pane shows the selected object.

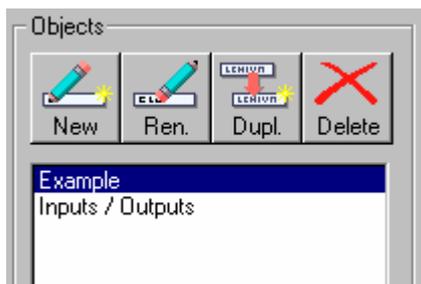
Object lists



You can create new lists, rename existing lists and delete a complete object list. The import function is used to [import standard object](#) delivered with an Fbox library or [Advanced Objects](#). Do not mix up your objects and the standard or advanced objects in the same list.

Select a list to view and [edit the object](#) contained in a list.

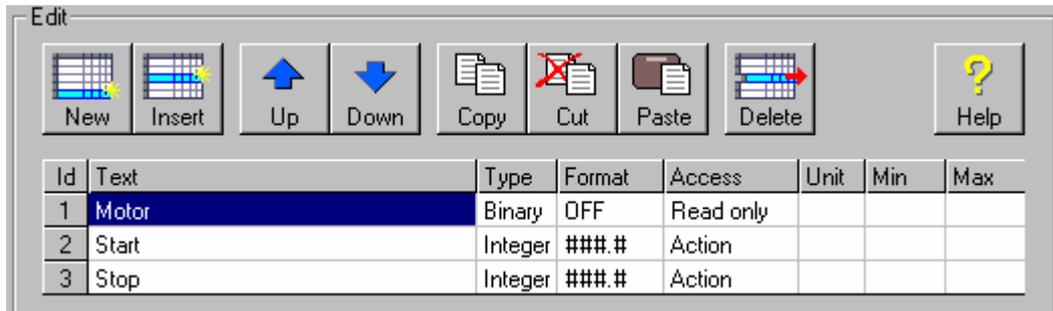
Objects



You can create a new object, rename an existing object and delete an object from the list. You may also duplicate and adapt an existing object rather than creating new objects.

The object names are only for user information and are not displayed on the terminal screen.

4.2 Edit objects



An object is made of 1 to 20 elements. To build an object you can insert new elements or copy an existing element. The Up and Down buttons allow you to arrange the elements inside the object.

An element is made of the following attributes.

- [Text](#)
- [Type attribute](#)
- [Format](#) (Format or Status)
- [Access](#)
- [Unit](#)
- [Min/Max](#)
- [Default Resources](#)

Select an attribute and press Enter, or Double click on an attribute to change the selected attribute. The corresponding Select tool is displayed.

You cannot modify Advanced Objects.

On a standard object you can only modify the following attributes:

- [Text](#)
- [Unit](#)
- [Min/Max](#)

You cannot modify the standard texts in the attribute tab but you can edit new texts and select them from a standard object.

It is also not possible to change the number of elements in a standard object.

Selecting attributes

Select Text	Select Format	Select Unit
Input 0	(none)	
Input 1	HH.MM Hour...	°C
Input 2	DD.MM Day ...	%
Input 3	DDW Day ...	V
Input 4	DD.MM Day ...	mV
Input 5	DDW Day ...	mA
Input 6	YY Year	W
Input 7	Up/Down Up /...	kW
Output 0	##### -999...	kWh
Output 1	###.# -99.9...	Hour
Output 2	##.## -9.99...	Sec
Output 3	#.### 0.00...	
Output 4	Message Mes...	
Output 5	Shor...	
Output 6	Long...	
Output 7	Comment	
	ON/OFF Binar...	
	Phone Pho...	

Inputs / Outputs Formats Status

In the Select Text, you may need to select the corresponding list to find your text. In the Select Format, you can toggle between the Format and Status selection. Remember that the status replaces the values and therefore no format can be defined at the same time as a status (except for the Up/Down format).

If the desired attribute is missing, you can use the Insert key to quickly add a new entry to the list. For better editing comfort, go to the [Object Attributes](#) tab.

4.3 Advanced Objects

Advanced Object is a feature that allows to import special predefined object in the HMI. An advanced object is not limited to strict structure of 1 to 20 element lines. The HMI library includes 2 default advanced objects supporting trend display for the D23x described below. Further advanced object can be developed on requests.

The default advanced objects are automatically installed with the HMI setup or the HMI Library installation. **Further advanced object must first be installed on the PC.**

The definition and the PCD code supporting an advanced object is not included in the basic kernel of the HMI. Therefore, advanced objects must be imported in the HMI file before to be used.

To import advanced objects in your HMI file, use the command 'Import Advanced Objects' in the 'File' menu or click the Import button in the Objects tab. Then select the folder where to look for the files. Default advanced objects are in the standard library folder while extra advanced objects are in the application library folder. When imported, a new list with the imported objects is created in the Objects tab. Such objects cannot be modified in the HMI Editor but you can use them like other objects in the menu tree.

If an updated advanced object library has been installed, you can re-import the new library in the same way. New objects will be added, modified objects will be updated. If objects should be removed from the list, you must remove the whole list and re-import it (individual objects cannot be removed).

Trend Advanced Objects

Two Advanced Objects are delivered with the HMI 1.3 to display trend diagrams for the D23x display.

- Trend positive
- Trend symmetrical

These 2 objects are similar. The first is designed to display positive values only, while the second support a symmetrical positive and negative range.

The user must provide a register array with the values to be displayed and a free PCD Text. The value range is free and must be specified by the user. Values out of the specified range will be truncated in the diagram. For the symmetrical trend, the range is specified in positive value but used symmetrically for the positive and negative ranges.

The displayed trend is a static diagram. It means, that the values are frozen when the object is called. To display new values in the diagram, the user can press Enter or recall the object with an F-Key (is defined for it).

Parameters of the Trend objects

Parameter	Type	Description
Address	R	Base address of a register array (size 25, 50 or 100)
Array size	K	Size of the register array
Range	K	Value range. The value corresponding to the max size of the Trend (taken as positive and negative for symmetrical trend).
Text	TEXT	Free PCD Text to be sent before the loading of the trend.

The register array must be of 25, 50 or 100. Other array size are not accepted. This parameter defines also the horizontal resolution of the trend. With 50 values, each value use 2 pixels and with 25 values, each one will occupy 4 pixels. The total width of the trend is always 100 pixels.

The height of the positive trend is 47 pixels, zero line included. The values are therefore internally converted from the user specified range into a range of 0 to 46. The symmetrical trend is also 47 pixel height, zero line included. The values are therefore internally converted from the user specified range into a range of -23 to +23.

A free PCD text must be provided to add text on the top line, for the 3 left characters of the lines 2 to 7 and the bottom line. The PCD Text must contain all necessary control characters to move the cursor through the lines 1 to 8. It is recommended to extend your text with spaces, where necessary, to overwrite the whole reserved place. The trend is always placed on lines 2 to 7, and use the columns 4 to 20.

Example:

```

"<26>** External Temp ** "
"Tmp<CR><LF>"
" <CR><LF>"
" <CR><LF>"
" <CR><LF>"
" <CR><LF>"
" 0 <CR><LF>"
"   Temp last 100h"

```

Tip : To collect the values in the register array, in IL, use the SHIU / SHID instructions and in S-Fup, use the FIFO Integer Fboxes from the Buffer family.

5 Status Bars tab

Go to the Status Bars tab to define the Status bars. The feature is only available if a terminal type supporting icons has been first selected (E.g. D23x). A total of 4 different bars can be defined for Root, Menus, Objects and Alarm buffer. The bars are always placed at the 2 bottom lines of the display.

To define a status bar, first select the type you want to edit with the top buttons.



Then check (or uncheck) the option to define that the selected bar must be shown. The bar will always include 6 icons. For each icons 3 working principle are possible.

Static icon.

To define a static icon, simply click the Icon button under the field of the corresponding position 1 to 6 and select a icon from the list. The list shows the icons you have [loaded in the HMI file icons](#). Leave the other field empty. A static icon is not very interesting unless it is combined with the blinking function described below.

Dynamic icon

A dynamic icon will be selected over a register by the user program. In this way, any icons loaded in the terminal can be displayed at a position. The register value (1 to 100) define the icon to be displayed. [See the icon list](#) to know and arrange the icon reference number.

To define a dynamic icon, click the Resource button below the corresponding position 1 to 6 and select a register from the resource list. Leave the other field empty.

Blinking icon

In most cases a blinking icon will be the more useful function. It allows to select between 2 icons by mean of a flag. To do so, click successively the 2 Icon button and select 2 static icons, one for the State 0, one for the state 1. Then click in the Selector field and select a flag from the resource list. The state 0 of the flag will display the icon 0 while the state 0 of the flag will display the icon 1.

You can also combine the dynamic icon function with the blinking function. To do so, you have to define 2 registers and a flag.

Note : It is not recommended to use Input as selector, since there state can change during the display of the icon. This may show different part (1st line and 2nd line) of an icon during short time.

6 Attributes

The object attributes are basic 'materials' to build the object elements.

On the top of the tab you can select the Attribute type you want to view and modify.

- Texts
- Status
- Units
- Min/Max
- Resources
- System Texts
- Icons

On the left pane you can select, rename and duplicate attribute lists. On the right pane, you can work on the selected list. Note that only Texts, Status, Resources and Icons use the lists. Other attributes are stored in a unique list.

To modify an attribute, select the item in the list (double click or Return). Note that some attributes are predefined and cannot be removed or modified.

Obsolete items can be removed with the delete button. However, you cannot delete an item currently used in a object. If required, delete or modify first the object.

6.1 Text attribute

Texts are grouped in several Text lists. These texts are used to label variables in objects. New texts can be defined in the [Object Attribute](#) tab. They are used when building objects in the [Objects](#) tab.

Note that the effective text length displayed on the screen is depending on the terminal type and sometimes the variable. Texts will be truncated by the compiler if necessary. You will see the effective text with the [Play feature](#) when the object is used in the menu.

6.2 Type attribute

The type attribute defines the value type in the PCD and activates the format conversion.

The following types are available:

Type	Description
Binary	The value is read from a flag.
Integer	For integer register. The value is read and written in a register as a normal integer value.
Float	For float register. The value is read and written in a register as a floating point value. Note that values are converted in integer for display and editing. They are converted back to float after modification.
Time	The value is read from a register as a time value in format HHMM. No special conversion.
Date	The value is read from a register as an integer representing a date in format MMDD (MM=month, DD=day). The value is converted in format DDMM for the display and for editing.
Hexa	For Hexadecimal values. Modem commands and phone number in the special Modem format used by the Fbox of the Modem library are also based on an hexadecimal format.

The types list is predefined and cannot be modified by the user.

6.3 Format attribute

The Format attribute defines the format used to display the value on the screen. You can choose a [status](#) in place of the format. This can be done by clicking the 'Status' button in the format selector. When a status is selected, the button is changed to '+Status'.

For the Up/Down and the Up/Off/Down formats, the Status is used in combination with the format.

For the ON/OFF, the Up/Down and the Up/Off/Down formats, if no status is selected, the default status are used. Default status can also be modified in the [System texts](#) in the [Object Attributes tab](#).

Each format is associated to default limits. The default limits are however not applied when the 2 lines option is selected because larger values can be displayed.

Format	Description	Default limits
HH.MM	Hour and Minute	00.00 - 23.59
DD.MM	Day and Month	01.01 - 31.12
DOW	Day of Week	1 - 7
DD.MM with 0	Day and Month with 00.00	00.00 - 31.12
DOW with 0	Day of Week with special functions	0 - 9
YY	Year	00 - 99
Up/Down	Up / Down commands (see detail below).	0 - 2 displayed OFF-UP-DOWN
#####	Numeric, no decimal point	-9999 to 99999
###.#	Numeric, 1 decimal digit	- 99.9 to 999.9
##.##	Numeric, 2 decimal digits	- 9.99 to 99.99
#.###	Numeric, 3 decimal digits	0 - 9.999
Message	Display of a text message (see detail below)	
	Short bar graph (see detail below)	0..100
	Long bar graph (see detail below)	0..100
Comment	Comment text only without variable	None
ON/OFF	Binary variable with status ON and OFF	0 and 1

Phone	Phone numbers	
Modem	Modem commands	
Up/Off/Down	Up / Down commands with automatic reset to Off (see detail below).	0 - 2 displayed OFF-UP-DOWN
Hexadecimal	Hexadecimal values, 8 digits	

The formats list is predefined and cannot be modified by the user.

Up/Down and Up/Off/Down format

These formats are used to give Up and Down commands on the terminal. Resource 1 defines the first of 2 flags (or Inputs/Outputs). The first element is set with the Up command. The second with the Down command.

On the keyboard, the Up command is set with the Up arrow key and reset with the Down arrow key. The Down command is set with the Down arrow key and reset with the Up arrow key.

With the Up/Off/Down format, both flags are automatically reset when the edition is ended. The default displayed statuses are OFF, UP, DOWN and UPDN if both elements are set. These statuses can be modified in the [System texts](#).

Message format

This format allows you to display a message text on the line. The resource 1 is a register containing the PCD text address. Therefore, you can display various messages depending on any condition. When the object is called, the text selected in the object is displayed. When the Enter key is depressed on the line, the message is displayed and starts to scroll. At the end of the text, the scrolling stops and the object text is displayed again. Any key pressed stops the message display during scrolling. The end of the text must be marked by a back-slash which is itself not displayed.

Example: "This is my message.\"

Bar graphs

These formats can only be used for terminals supporting the bar graph function (e.g. D23x). The short bar graph will use the same place as other numeric variables format. The long bar graph however is extended and the text is shorted to 4 characters only. The valid value range for bargraph is 0 to 100. Larger values may disturb the display. The default limit is set to this range. The user should not define a larger range. When the value is defined by the PCD, the user program should ensure that the value will not be out of the range 0..100.

6.4 Status attribute

The status attribute can be selected to replace a numeric value on the screen. A status is a text of one to four characters. New status texts can be defined in the [Object Attribute](#) tab. They are used when building objects in the [Objects](#) tab.

Some status lists are predefined and cannot be modified. You are free to add new status lists. Up to 100 lists can be defined and each list can contain up to 100 status.

The first status of the list is associated to the value 0, the second one, to the value 1, and so on. Therefore you must enter your status in the correct order. You must also create the user program producing numbers corresponding to the status. The user program should ensure that the values start with 0 and don't overpass the defined status list. If the value is bigger than the last existing status, the last is displayed. This can be used to display an error status for all invalid values. When a value with status can be edited, you should also define min/max limits to avoid invalid values to be entered by the user.

Example:

A register must display some process status like:

- Initializing
- Ready
- Running
- Ended
- Error

The following status list can be entered and associated to the numeric values:

- INIT 0
- RDY 1
- RUN 2
- END 3
- ERR 4

Select this status list in place of a format in the object at the corresponding element line. The program should also ensure that no other values can appear in the register.

6.5 Access attribute

The access attribute defines the mechanism associated to the element that can be applied by the user on the terminal. For each access mechanism a corresponding sign is displayed on the last column of the screen. Some access mechanisms need an auxiliary flag.

Access	Description	Sign
Read only	Editing of the variable is not allowed.	(none)
Read/Write	Editing of the variable is allowed.	<
Override	Editing and overriding allowed. The auxiliary flag is set when the value is modified. The flag can be reset by the user with the Acknowledge key (label Q or Quit). See example below.	= and #
Update	Editing allowed. The auxiliary flag set when the value is modified. See example below.	<
Action	Immediate execution on Enter key. The auxiliary flag (Resource 2) is immediately set when the Enter key is depressed. The Resource 1 can be an Integer register or a flag to be displayed. For instance a running timer or an ON-OFF state. See example below.	!
Edit enable	Editing is allowed if the auxiliary flag (Resource 2) is set. This access type can be used in combination with the password mechanism. It is intended to avoid editing elements depending on the status of the process.	<

It is the task of the user program to execute interactions between the auxiliary flag, the variable and other code and variables of the application.

Code example for an Action:

```

STH    Resource2    ;;Detect action
RES    Resource2    ;;Reset action flag
LDL    T 0          ;;Start timer
        100         ;;To 10 sec
COPY   T 0          ;;
        Resource1    ;;Display running timer

```

Code example for an Update:

```
STH    Resource2    ;;Check update flag
RES    Resource2    ;;Reset update flag
JR     L skip       ;;Skip update
COPY   Resource1    ;;Save the new value
        R 0
IFP    R 0          ;;Update the float value
        0
```

Skip:

Code example for an Override:

```
STH    Resource2    ;;Check override
JR     H skip       ;;Yes, skip refresh
STH    I 0          ;;
OUT    Resource1    ;;Image of input 0
```

Skip:

Code example for an Override Integer:

```
STH    Resource2    ;;Check override
JR     H skip       ;;Yes, skip refresh
COPY   R 0          ;;Copy input
        Resource1    ;;Image of R 0
```

Skip:

6.6 Unit attribute

Units are only available for large terminals (PCD7.D250) and for small terminals (PCD7.D23x) when the option '2 Lines' is selected.. A unit is a small text of up to 4 characters placed after the value.

Some default units are predefined. You are free to modify and add new units to the list. The list can contain up to 100 entries.

Go to the [Object Attributes](#) tab to define new Units. Units are used when building the objects in the [Objects](#) tab.

6.7 Min/Max attribute

For modifiable values, you can define value limits. If no limits is selected, default limits will apply. Limits are not defined individually to each element. Limits can be defined once and used on several elements. A set of limits is made of a Minimum and a Maximum value. Therefore, when an object is built, it will automatically have its associated limits. You should prepare limits according to the kind of parameter that will apply to. Therefore, when you modify a limit all associated elements will use the new value.

Important ! You must take care at the value format of the limits and the variables they apply to. Don't mix up the variable format and the display format !

Example : If the limit is used for a floating point variable, the value must have a dot even if there is no significant digit after it (E.g. 123.4 or -99.9 or 50.0). If the limit is used for an integer you must omit the decimal point even if the value is displayed with a decimal point (E.g. 123 and 9999 for 12.3 to 999.9 on an integer display in format ###.#).

Some limits are predefined. You are free to modify and add new limits to the list. The list can contain up to 100 entries.

Go to the [Object Attributes](#) tab to define new limits. Limits are used when building the objects in the [Objects](#) tab.

6.8 Default Resources

When you define an object, you can already define default symbols for the resource 1 and 2. Use the horizontal scrollbar at the bottom of the window if necessary to display the 2 Default Resources fields.

These symbols should contain a common group part. The subject is then to quickly initialize the default resource to an object instance and to rename all resources at once.

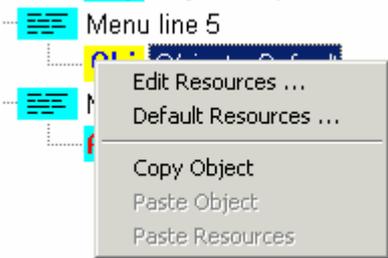
Id	Format	Access	Unit	Min	Max	Default Resource 1	Default Resource 2
1	###.##	Read/Write/	°C	-99.99	99.99	default.setpoint	default.enable
2	HH.MM	Read only	h/mn			default.display	

To use the default resources proceed as follow :

Call the object as usual in the menu tree.

Right-click on the object line in the menu tree.

Choose the command Default Resources.



All resources will be set to the default and the Resource edition window in open to allows you to give a new group name to all resources.



To quickly change all resources again at later time, use the Edit Resources command in the same menu.

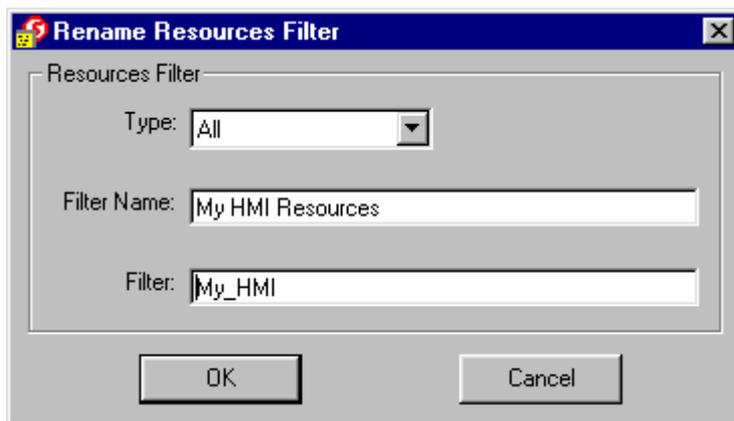
If the resources does not have the same group name, only the ones matching the selected resource are renamed. Resources without group name cannot be renamed with this feature.

6.9 Resources

Go to the [Object Attributes tab](#) to view the available resources and to define resource filters. Resources are not associated to the objects in the objects lists. Each object instance will use its own resources. Therefore you must associate resources to objects that are called in the [menu structure](#).

Flags and registers (Integer and Float) are always supported in a HMI application. You can also use other resources (Inputs / Outputs / Timers and Counters) as long as the target system has a recent firmware. To do this, first select the option 'All media types' in the [Settings tab](#). The minimum FW versions are given in the description of this option.

The resources are placed in a single resource list but can be filter by definable filters. Use the filters to group resources according to your favorite categories.



On the left side, you can define new filters (New button) to get a better overview of the interesting resources, rename and change existing filters (Edit button) or duplicate and modify existing filters (Dupl. Button).

The filter can define a resource type and the first matching characters of the symbol. The resource list contains all symbols but only the ones matching the selected filter are displayed. The same resource can appear with several filters.

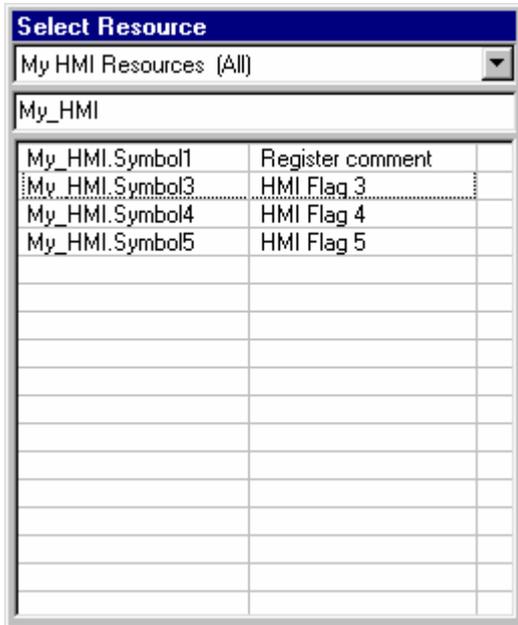
At startup of the HMI, the Global list is automatically loaded and the changed resources are updated (type and comment).

You should always change symbols and introduce new symbols in the Global list of the Project Manager or from an editor like S-Edit or S-Fup. When you define a new global symbol, it is not immediately available in the HMI-Editor. You must save the symbol in the Global symbol file as follow :

- If you have introduced a symbol in Project Manager, close the resource list and re-open it.

- If you have introduced a symbol in an Editor, save the file of this editor.
Now use the 'Reload Global' command with the Actions button in the HMI-Editor.

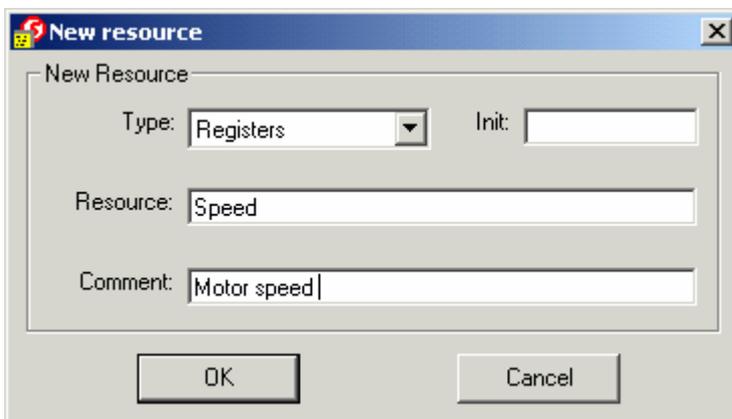
The filters are very useful when you have to select resources in object instances.



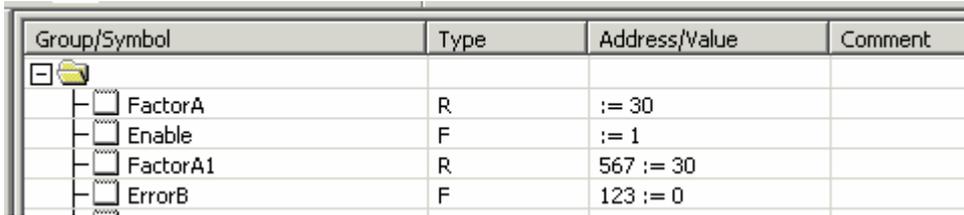
The select resource tool has tree parts. The top line allows you to select a defined filter. The second line allows you to extend or totally change the current symbol selection. The last part lists the matching symbol. Once, you have fined your resource, select it in the list and press Enter or select it with the mouse.

Init value

Resources can optionally have an initial value. The variable will be automatically initialized at startup after a program download (not at each power up). Take care that the format of the registers is correct. The initial value must have a dot to be considered as a floating point format.



Important note: In PG5 1.2 and up, init values should be defined in the Global list and not anymore in the HMI. To do this, use the following syntax:



The screenshot shows a software interface with a table. The table has four columns: 'Group/Symbol', 'Type', 'Address/Value', and 'Comment'. The 'Group/Symbol' column contains a tree view with a folder icon and four sub-items: 'FactorA', 'Enable', 'FactorA1', and 'ErrorB'. The 'Type' column contains 'R', 'F', 'R', and 'F' respectively. The 'Address/Value' column contains ':= 30', ':= 1', '567 := 30', and '123 := 0' respectively. The 'Comment' column is empty for all rows.

Group/Symbol	Type	Address/Value	Comment
FactorA	R	:= 30	
Enable	F	:= 1	
FactorA1	R	567 := 30	
ErrorB	F	123 := 0	

Conversion of old HMI files

Because of compatibility with old HMI files, resources that are not from the Global list may also be in the list. You can still introduce resources manually in the list.

If manually introduced symbols are used, they must also be declared in the global list. When the HMI file is open, these symbols will be updated with the current global list.

If manually introduced symbols are not used, they should be removed by using the command 'Remove unused' with the Action button.

6.10 System texts

The list of system texts is available from the [Object Attribute tab](#). Here, you can modify some default texts used by the HMI. The users don't have to care about the length of the text. The HMI compiler will cut or extend the text to the correct length. For long texts, depending on the terminal size, if the 2 first characters are identical, it will be used to extend the text. The same is valid for the 2 last characters. In this way, the text in between will be automatically center on the display.

Name	Default	Length	Description
Alarm buffer A Top	--[Top]--	Var	Top mark of alarm buffer A
Alarm buffer A End	--[End]--	Var	End mark of alarm buffer A
Alarm buffer B Top	--[Top B]--	Var	For future use
Alarm buffer B End	--[End B]--	Var	For future use
Alarm High	ALA	4	Alarm High status
Alarm Low	OK	4	Alarm Low status
Alarm Not Acknowledged	NAK	4	Not Acknowledged status
Alarm Acknowledged	OK	4	Acknowledged status
General Alarm High	ALA	4	Alarm status for the general alarm to be displayed in the root menu
General Alarm Low	OK	4	OK status for the general alarm to be displayed in the root menu
General Alarm Not Ack	NAK	4	Not Acknowledged status for the general alarm to be displayed in the root menu
General Alarm Ack	OK	4	Acknowledged status for the general alarm to be displayed in the root menu
Binary OFF	OFF	4	OFF state of binary signals.
Binary ON	ON	4	ON state of binary signals.
Duplex OFF	OFF	4	OFF state of duplex signals (Up/Down format).
Duplex Up	UP	4	UP state of duplex signals.
Duplex Down	DOWN	4	DOWN state of duplex signals.
Duplex Up+Down	UPDN	4	UP+DOWN state of duplex signals (should not occur).

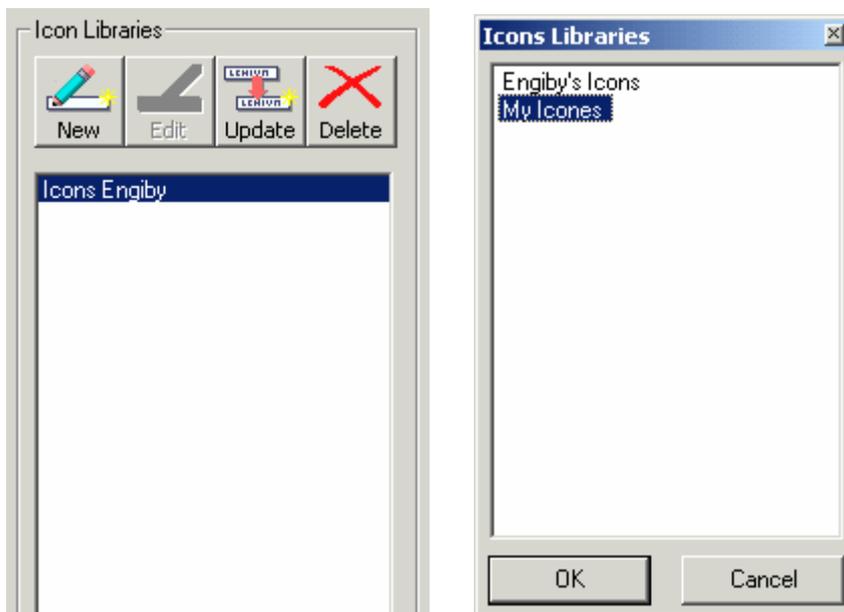
Length=Var means variable length depending on the terminal size.

6.11 Icons

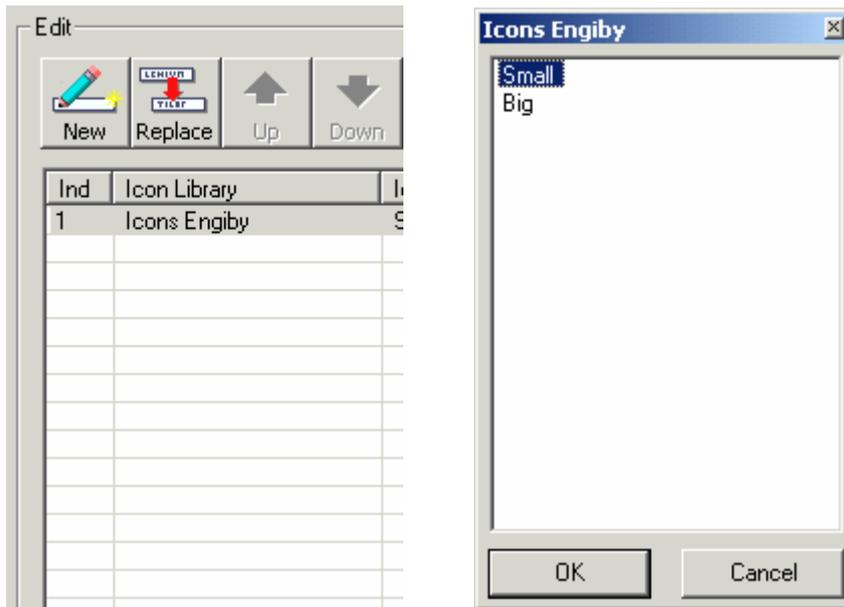
Go to the [Object Attributes tab](#) to view and select the icons in your HMI file. Icons are edited with the Icon Editor and saved in a central location on your PC. You can also receive Icon Libraries from SAIA or third party.

With the HMI-Editor, you can load up to 100 icons in your HMI file as follow.

You must first select the icon libraries from which you want to use icons. Click the New button of the Icon Library and select a library. Icons are not yet loaded but the library is now known in the HMI file.



To load icons in the HMI file, select a library in the Icon Library list and click the new button in the Edit frame. If you want to load icons from another library, always first select the Icon Library in the list and then use the New button from the Edit frame.



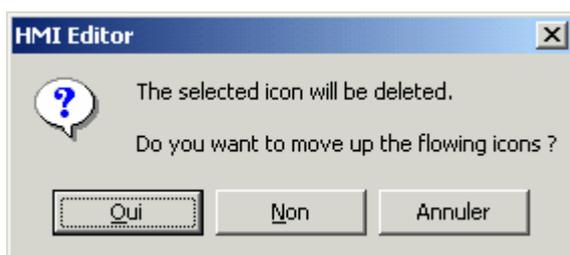
The Update button must be used to reload in the HMI file, icons that have been modified by the Icon Editor or if you have received an updated Icon Library.

The icons are numbered from 1 to 100. This numbering is important when you want to select icons for the status bar over registers. The value in the register define the icon to be displayed. You may also use the Up and Down arrow buttons to arrange the icon list according to the values you will find in the register.

The Delete button of Icon Library frame, remove an Icon Library from the HMI file but do not delete the Icon file on the PC.

The Replace button of the Edit frame allows you to select an new Icon for the selected position. Only icons from the same library can be selected like this. To replace the icon by a new one from another library, you must first delete it.

The Delete button from the Edit frame, allows you to remove an icon from the HMI file. Doing this, you will be prompted to decide if you want to move up the following icons or not. Take care that moving up the following icons will modify there reference number and may alter the functioning of a status bar selecting icons over registers.



Load icons in the terminal.

The icons must be loaded in the terminal at each power up of the PCD and also after a power up of the terminal only. This time is depending on the number of defined icons. It can take up to 10 sec for a list of 100 icons. During this time, the Root menu is already displayed but icons may be missing in the [status bar](#). It is also not possible to enter a password and to access the menus until all icons are loaded.

This mechanism can also be restarted if the serial connection is interrupted.

7 Alarms

7.1 Alarms tab

The Alarms tab allows you to create a list of alarms, associate the alarm flags and define the format of the alarm message in the buffer.

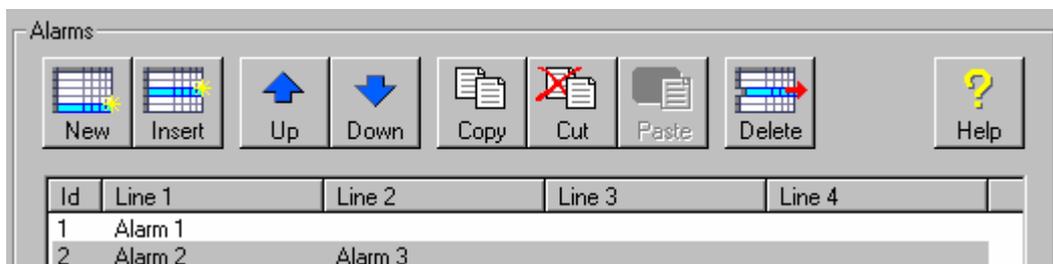
Buffer settings



A checkbox allows you to enable or disable the whole alarm function. The Settings button allows you to adjust general [alarm settings](#).

Note that only one alarm buffer (Buffer A) is support in this version.

Alarm list



The alarm list shows all defined alarms and allows you to add, remove and arrange the alarm list. Select an alarm to edit the alarm properties below.

Alarm properties

	View for D160 with 4 * 16
Line 1	Alarm 1
Line 2	Alarm 2
Line 3	Alarm 3
Stamp	21.07 15:03:50
Status	ALA NAK 0024

You may define up to 4 text lines for each alarm and add two optional lines for a time stamp and status information. The status line shows the alarm status (ALA or OK), the acknowledgement status (NAK or OK) and a counter indicating the number of alarm events.

	Alarm flag	NAK flag
Resources	F 100	

In the resource fields, you can define the alarm flag and the NAK flag. The alarm state can be defined as 1 or 0 by an option in [alarm settings](#). If no alarm flag is defined, the alarm function is disabled but the entry remains in the alarm list.

When an alarm arises:

- The alarm is searched for in the buffer
- If the alarm is not yet in the buffer a new record is created
- The time stamp is updated
- The alarm state is set to ALA
- The acknowledgement state is set to NAK
- The counter is incremented

As long as the alarm is active, the General alarm flag is also set. See [Internal tab.](#)

The NAK (Not acknowledged) flag is optional. If no flag is defined, a internal flag will be used and then the alarm can only be acknowledged from the terminal in the alarm buffer. The NAK Flag is set when an alarm arises. It will be reset by an acknowledgement on the terminal. The alarm can also be acknowledged by the user program on the NAK flag. The NAK state can be defined as 1 or 0 by an option in the [alarm settings](#).

Finally, you should define a NAK flag if you want to acknowledge the alarm from the user program or if you want to know the NAK state in the user program.

As long as the alarm is not acknowledged, the General NAK flag is also set. All alarms are acknowledged and all NAK flags are reset when the General ACK flag is set. See [Internal tab.](#)

7.2 Alarm settings

The alarm settings are available from the [Alarm tab](#).

The following properties are available:

Buffer length	Define the maximum number of alarms recorded in the buffer. The number of lines used by each alarm does not affect this amount. See below for length > 50.
Auto display	Define if a rising alarm will interrupt the current view and automatically display the alarm buffer.
Close when empty	Define if the alarm buffer must be automatically closed when the last alarm is removed. The option 'On timeout' allows you to close the alarm buffer only if a timeout occurs (when the standby menu is called and password is cleared).
Buffer mark	Define if the marks 'Top' and / or 'End' must be shown in the buffer.
Buffer full	Define the handling of a new alarm when the buffer is already full. 'Shift' will shift the buffer content and insert the new one. 'Ignore' will discard the new alarm.
Insert new alarm	Define if a new alarm is inserted on the top or at the end of the buffer
Acknowledgement	Define if the Enter key immediately acknowledges an alarm or if the acknowledgement key (labeled Q or Quit) is required after the Enter key.
Automatic alarm removal	Define if an alarm which is reset must be automatically removed from the buffer. The option 'On ACK' allows you to remove the alarm only if it is also acknowledged.
Manual alarm removal	Define if alarms can be manually removed from the buffer. The option 'If reset' allows you to manually remove alarms that are already reset (not anymore in the alarm state).
Move cursor down	Defines if the down key will jump the cursor line per lines or screen by screen. 'Screen' means all lines of an alarm. Restrictions with option "by screen": 1 - The cursor cannot be moved to the status line (ALA – NAK) and the alarms cannot be acknowledged on the display. 2 - If the alarms (texts+ status lines) are longer than the display, the last line(s) of the alarm record cannot be

displayed.

Note that the up key always moves screen by screen.

Alarm state	Defines which state of the alarm flag (High or Low) will be the alarm state. The opposite is consequently the OK state.
NAK state	Defines which state of the NAK flag (High or Low) will be the 'Not Acknowledged' state. The opposite is consequently the OK state (acknowledged).
Store all events	<p>This option cause all rising alarms to be separately stored in the buffer with its own timestamp and status. Also each event must be acknowledged individually.</p> <p>Despite that several events of the same alarm can be stored in the buffer only one NAK flag exist. Therefore, when the alarm is acknowledged on the NAK flag, all events in the buffer are acknowledged. When any of the event is acknowledged on the terminal, the NAK flag is acknowledged.</p>

Alarm buffer length and PCD memory

By default, the buffer length is set to 50. This buffer size doesn't require special configuration or equipment in the PCD. For performance reasons, it is strongly recommended to set the RAM DB above 3999 in the dynamic resource settings in the project manager.

For buffer size above 50, you must respect the following conditions:

- You must have equipped and configured an extension memory in your PCD system.
- You must have set the RAM DB range from 4000 or above in the dynamic resource settings in the Project Manager.

The same conditions apply, also for small buffers, if you have set an EPROM or a Flash memory as user program memory.

Automatic alarm removal options

No	Alarms will not be automatically removed.
If ACK	Acknowledged alarms will automatically be removed. The alarm state is not relevant.
If Reset	Reset alarms will automatically be removed (not anymore in alarm state). The acknowledgement is not relevant.
If ACK+Reset	Alarms that are acknowledged and reset will be automatically removed.

Manual alarm removal options

No	Alarms can not be manually removed.
Always	Alarms can always be manually removed.
If ACK	Only acknowledged alarms can be removed. The alarm state is not relevant.
If Reset	Only reset alarms can be removed (not in alarm state). The acknowledgement is not relevant.
If ACK+Reset	Only alarm that are reset (not in alarm state) and acknowledged can be manually removed.

Working in the alarm buffer on terminal.

The keys Up and Down allows you to navigate through the alarm buffer.

To acknowledge an alarm, select the status line, and press Enter. If the option 'Enter+Quit' is set, a star is shown at cursor position when you press the Enter key. Then you can abort with the Escape key or acknowledge with the Quit key.

To completely remove an alarm from the buffer, select the first text line and press Enter. A star is shown at cursor position. Then you can abort with the Escape key or remove the alarm with the Quit key. This action is depending on the option 'Manual alarm removal'.

For a terminal without 'Quit' key, you must press two times the enter key. This is the case for instance for the D230, the D790 and the D795 terminals.

Tips

- It would be a good practice to define a F-key to quickly access the alarm buffer.
- To improve the readability of the alarm list, it would be good to have a visible mark as first line.

Example : ===== Alarm =====.

8 Settings

8.1 Settings tab

The Settings tab contains general properties of the HMI project grouped in:

- [Terminal](#)
- [Options](#)
- [Serial line](#)
- [Terminals list](#) (for bus topology only)
- [Passwords](#) (with the password button)

8.2 Terminal settings

This frame collects the terminal properties:

- Terminal type:** Select the target terminal. This setting will select the specific terminal commands and key codes for the PCD.
- Terminal topology:** Define if on or more terminals will be connected to the PCD and the used topology. Only the topologies supported by the selected terminal are shown. If the multipoint or the bus topology is used, the serial line selection or the terminal list is enabled and must be filled out. See also the topic [Topology](#).
- Terminal(s) number:** Activate only with the multipoint topology. Depending on this option, the corresponding number of serial lines are activated and can be defined.
- Play as:** All terminals are not known by the HMI Editor. However, the HMI Editor is able to preview your work on a similar terminal selected here. This option affects the Object preview on the Menu Structure tab and the Play feature.
- Allow setup:** The setup access is enabled or disabled on the terminal. Valid only for terminals supporting this feature.
- Character set:** Select the character set used by the terminal. Valid only for terminals supporting this feature. If not supported, select the Default option. You must be sure that the terminal being used supports the selected character set.

Note also that the supported special characters depend on the used terminal and the selected character set. Refer to the manual of the terminal and check the character table supported for each language. Some very special characters like Greek letters are also not supported.

The 'Default settings...' button allows you to reset the terminal settings and the serial line settings to default values according to the selected terminal.

8.3 Topology

The used topology, the serial lines and the terminal addresses are defined in the [Settings tab](#).

Tree topologies are available:

- [Single terminal](#)
- [Multipoint terminals](#)
- [Bus terminals](#)

#Single terminal

The 'Single terminal' topology allows you to connect only one terminal. Only one serial line is used and must be configured.

In the root menu, only the Login menu is available and can be defined.

At start up of the PCD, the Login menu is automatically displayed. The password can be entered to login on the terminal.

#Multipoint terminals

The 'Multipoint' topology allows you to connect up to 6 terminals of the same type to one PCD. Note that only the PCD2.M170 supports 6 lines and the PG5 version 1.1.xxx is required for the line 5. The number of connected terminal must also be selected and the corresponding number of serial lines must be configured using the buttons 'Terminal x'.

In the root menu, the Login and the Busy menus are available and can be defined. In this case, variables can also be displayed on the busy menu because it will be cyclically refreshed.

Only one terminal can be active at the same time. At start up of the PCD, the Login menu is automatically displayed. The password can be entered to login on the terminal. At the same time, the other terminals are set in a busy state and the Busy menu is displayed until the used terminal is set back to the Login menu or the timeout is elapsed.

#Bus terminals

The 'Bus' topology allows you to connect up to 31 terminals of the same type on a single RS485 bus. This option is only available for terminals supporting a bus protocol. When this topology is selected, the terminal list is automatically activated and must be filled out with the connected terminals.

In the root menu, the standby, the Login and the Busy menus are available and can be defined. In this case, variables cannot be displayed on the busy menu because it will not be refreshed.

Only one terminal can be active at the same time. At start up of the PCD, the standby menu is display on each terminals. When any key is depressed on a terminal, the Login menu is displayed. The password can be entered to login on the terminal. At the same time, the other terminals are set in a busy state and the Busy menu is displayed until the used terminal is set back to the Standby menu or the timeout is elapsed.

8.4 Serial line settings

This frame collects the serial line properties:

Serial line:	Serial line on the PCD connected to the terminal(s). Be sure the line is not configured as S-Bus PGU.
PGU line:	Select Yes if the line is also used as PGU line (e.g. line 0 or line 4 on PCD6).
Type:	Type of the selected serial line. Be sure the correct module is plugged in to the PCD. See the terminal and the PCD manual for cabling. See remark below for terminal D79x.
Baudrate:	Select the used baudrate. The same selection must be done on the terminal. See remark below for terminal D79x.
Bits-Parity-Stop:	Select the used values. The same selection must be done on the terminal.
Handshake:	Select the used handshake. If the selected speed is too fast for the terminal, it is recommended to use a reduced baudrate. As security, to prevent terminal buffer overflow, a handshake can be used. If hardware handshake is used, the corresponding control lines must be connected.

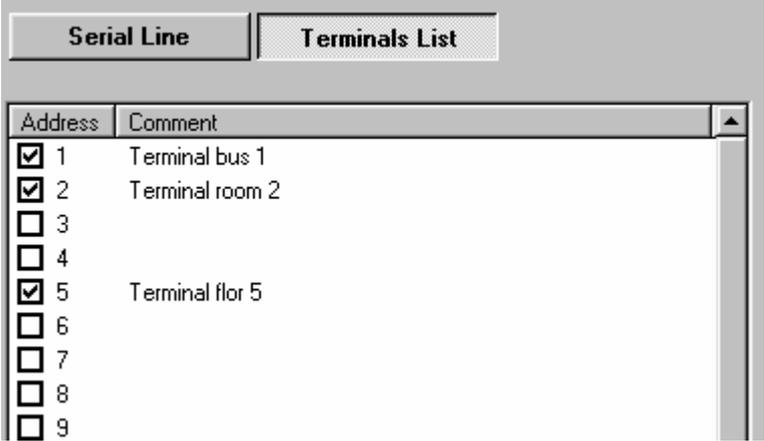
Remark for terminals D79x.

The maximum speed allowed for these terminals, used with the bus topology (RS485) is 9600 bauds.

Problems can occur on RS485 because the timing supported by some PCD is not optimized for these terminals. In case of problem, a terminal will start login itself and wait for the password. Other terminals are busy. In this case, you should use the special Type RS 485 Short timing to solve the problem. This uses the MC5 mode supported by new PCD firmware. Please consult the corresponding documents for the list of FW supporting this communication mode.

8.5 Terminals list

The terminal list is available in the [Settings tab](#) if a terminal type supporting a bus protocol is selected and the bus topology is set.



The list is shown by clicking the 'Terminal list' button. Each connected terminal must be selected. An optional comment can be edited for each terminal.

8.6 Settings - Options

This frame collects various menu and object options:

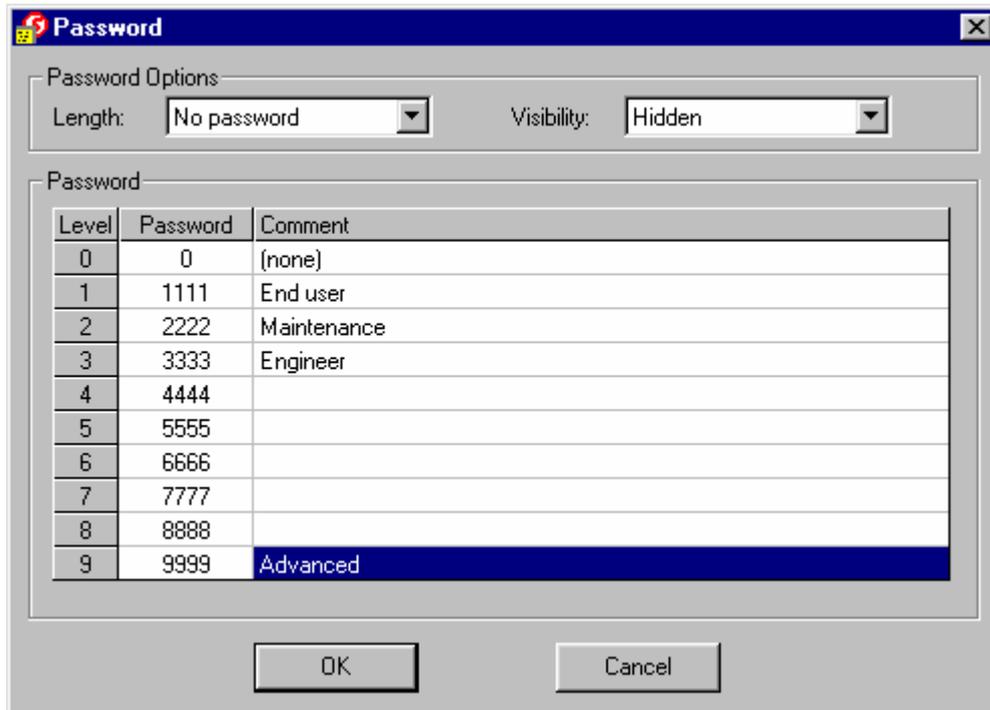
- Timeout [hh:mm:ss]:** If no key is pressed during the specified time, the screen is switched off, the password is cleared and the root menu is displayed. The value is limited between 10 sec and 23h, 59min, 59 sec.
- Title in object:** If selected, the calling menu text is displayed as a fixed title at the top of objects.
- Lines per elements in object:** On small terminals, you may use 2 lines per element in objects. In this case, a full line is available for the text and more space is available for values. The status will also be shown.
- Supports all media types:** If this option is set to No, you are only allowed to use flag and Registers as variables in objects. accepting this restriction allow you to run the application on any firmware (except very old ones). If the option is set to Yes, you are allowed to directly specify all media types (Input, Output Flag, Register, timer and Counter) as variables in objects. However you must dispose at least of the following firmware and CPU :

CPU	Official FW	\$ or Beta FW
PCS1.C8xx	096	B93
PCD1.M1xx	080	B7I
PCD2.M11x	090	\$8B
PCD2.M12x	090	\$8B
PCD2.M15x	0C0	BBJ
PCD2.M17x	010	
PCD2.M48x *	010	\$04
PCD3.Mxxxx	010	\$01
PCD4.Mxx0	Not supported	
PCD4.M1x5	0F0	BE3
PCD4.M445	0F0	BE3
PCD4.M17x	010	
PCD6.M54x	Not supported	
PCD6.M1xx	Not supported	
PCD6.M2xx	Not supported	
PCD6.M3xx	040	B37
PCD2.M9xx	Not yet supported	

* PCD2.M48x only from HMI Lib \$ 1.2.059

8.7 Password settings

Click the Password button in the [Settings tab](#) to define the passwords and the passwords options.



Password length option:

- | | |
|------------------|--|
| No password | No password is required. Pressing Enter will always allow login. The list is ignored. You may also use this option to disable all access restrictions set in the menu structure. |
| 4 digits | The passwords have a fix length of 4 digits. It can be edited with the Up and Down keys. Each digit must be confirmed with the Enter key. This option should be used with terminals without numeric keys. To make the entered password to the user, choose the format 'Visible' for the password display in the root menu. |
| 8 digits | Same as the '4 digits option' but with 8 digits passwords. |
| Variables length | The passwords can have different length from 1 to 8 digits. All keys are used as numeric digit. The Enter key ends the password edition and checks its validity. Only useful on terminals with numeric keys. |

Visibility option

Hidden	The password edition is not visible. Stars are displayed instead.
Visible	The password is visible during edition.
Edit visible	Only the edited digit is visible. Other digits are displayed with star instead. This option is useful for terminals without numeric keypad. The user see the digit he is editing with the Up/Down arrow keys or the rotary button.

The level 0 give less rights while the level 9 gives all rights. Ten default passwords are predefined. You may change any password. The valid range is from 0 to 99999999 (eight digits).

The entered password is evaluated as a numeric value. Therefore, leading 0s are ignored and the password 0 allows you a direct access by pressing the Enter key. The passwords are searched in the list from position 0 to position 9. Therefore, if the same password is defined twice, the lower level is activated.

The required password levels and restrictions are defined in the [Menu structure](#) for each menu position.

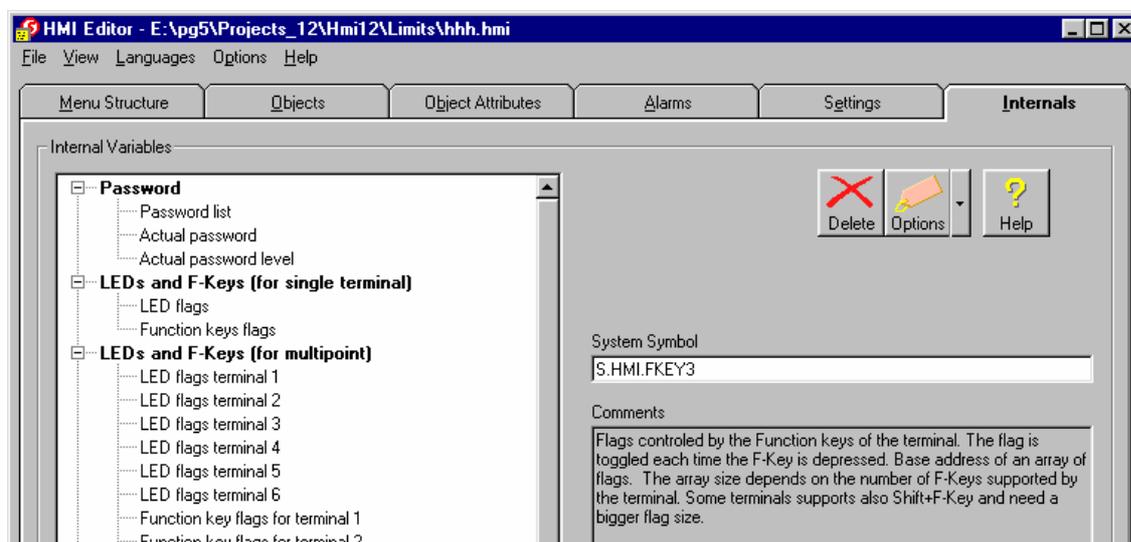
System symbols for the password list, the actual password and the actual level are available. See the list of [internal resources](#).

9 Internal

9.1 Internal tab

This tab shows the internal variables that can be accessed by the user program to interact with internal functions of the HMI. The internal variables can be accessed by the predefined System symbols. The system symbols is a new concept introduced in PG5 to allow the user program to access variables defined by the Program Manager, Editors and Configurators or Fboxes.

All available symbols are listed and described in the Internal tab. The symbols can be directly used in S-Edit or S-Fup module but also in the HMI-Editor itself.



Special case : The free menu texts

The free menu texts is a special case. In this tab, you cannot enter the content of a text. You need to define the text symbol and its content in the global symbol list. The reference to these text must be given in the Root Menu edition window (Menu Structure Tab). The field to enter the text symbol, will appear as soon as you select the free menu option.

Import of old HMI files.

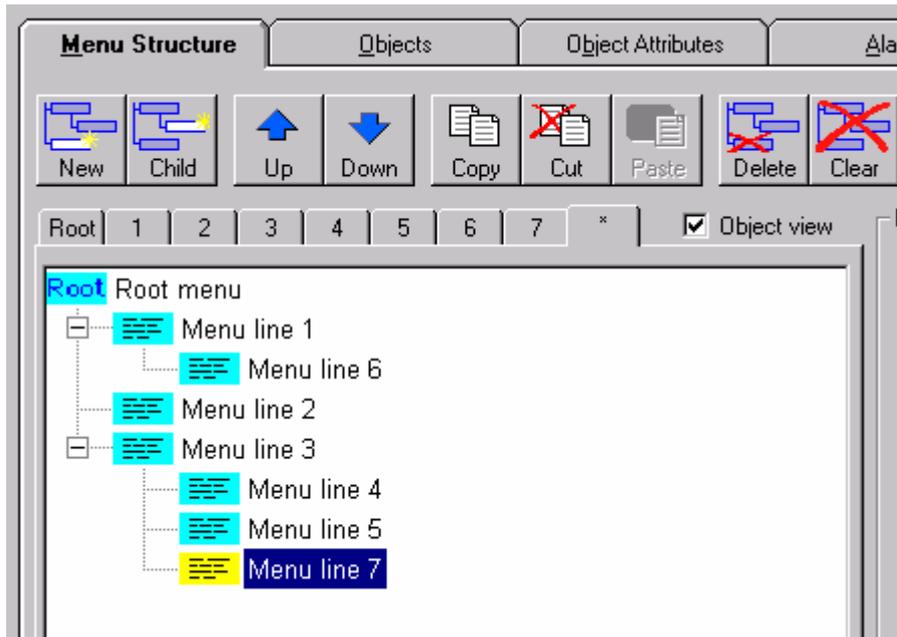
With this new concept, there is no need to define a variable address. But if, in a former version, you have defined addresses, they will be displayed and you can see and modify them. However it is recommended to remove these address and to use the predefined symbol in the future.

If you have included symbol definitions files you can still access it with the Option button. But it is recommended to remove such include files and to move these declaration in the global symbol list.

10 HOW TO ?

10.1 Build the menu structure

Starting from the root menu, add new menu positions using the New button. You can rename each menu position to suite your application. The text entered in the menu tree, is the text displayed on the terminal. Remember that the screen has a fixed size and your texts will be truncated accordingly.



To preview the created menu, use the [Play feature](#).

10.2 Create the PCD program

To create the PCD program click the Compile button on the [Menu Structure](#) tab. The compiled file will be created in the current folder. This file must be assembled and linked with the project.

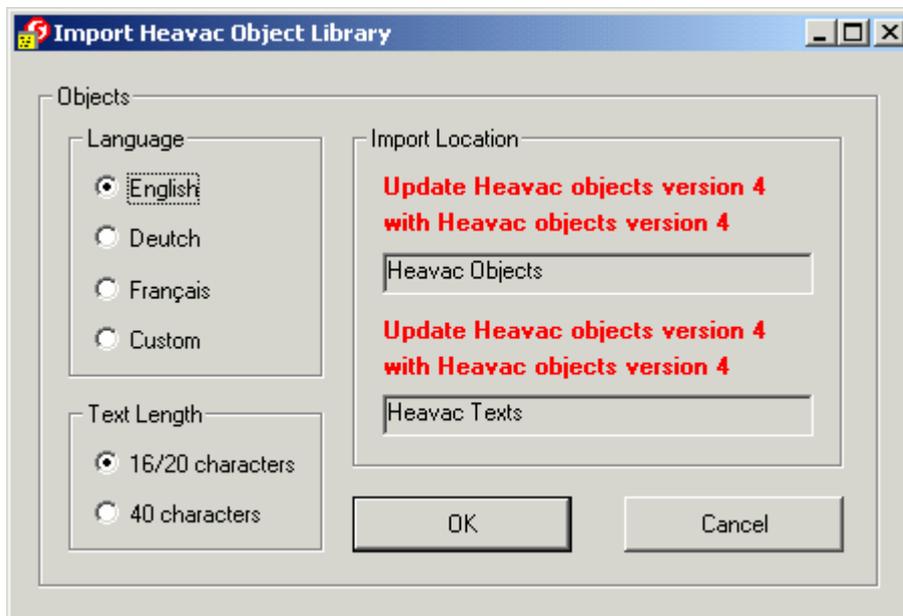
Attention ! The file is not automatically saved and compiled when you use the 'Build' or the 'Rebuild all' command from the Project Manager.

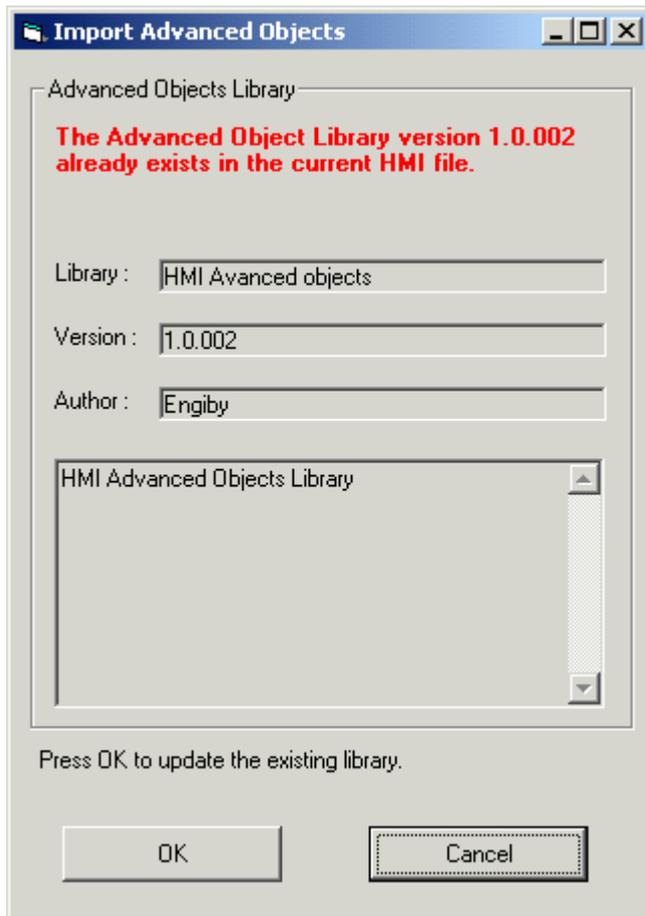
10.3 Import standard objects

This description is also valid for the import of [Advanced Objects](#).

Standard objects are delivered with the corresponding Fbox library. If the library is installed, you can import the standard objects into your project.

In the object tab, use the import button of the Lists toolbar to import standard or advanced objects. Then select the folder to open (Standard or Application). Select the file you want to import and confirm.





The import dialogue box allows you to select the language and the text length to suit your terminal. The name of the objects list and texts list created are displayed for information.

You may re-import the objects at any time in order to change the text language or length. You will also need to re-import the standard objects when new ones are added in the library.

You are not allowed to modify the structure of imported objects and edit imported texts. However some attributes of objects can be edited in the [Edit object pane](#).

See also the following topics to know how to use standard and advanced objects:

- [Call TypeUse standard objects](#)
- [Advanced Objects](#)

10.4 Use standard objects

Standard objects are delivered with a Fbox library and allow to access internal Fbox parameters from the terminal.

You must have installed a Fbox library including standard HMI objects first. If this is correctly done, you will have a LMI-file stored in your Fbox folder.

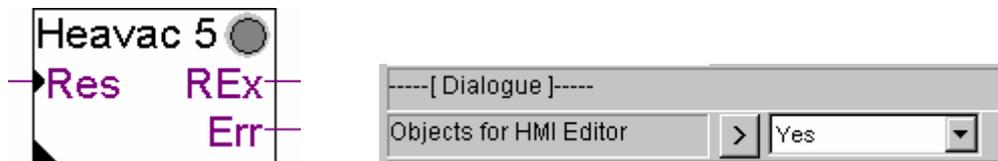
The Heavac library includes HMI objects since version \$ 2.1.259 from 31/05/2000.

The Modem library includes HMI objects since version \$ 2.3.008 from 02/10/2002.

You must [import the standard objects](#) in your HMI file using the Import function of the [Objects tab](#).

In the S-Fup file, you can use the Fbox supporting the HMI object. For Heavac Fboxes, activate the HMI option in the adjust window. Give a unique name to your Fbox by using the Fbox properties command in the context menu.

Important : For the Heavac library, use the **Heavac Init 5** Fbox and activate the option 'Object for HMI Editor' in the adjust window of the Heavac Init 5 Fbox. For the Modem library, use the **Modem Driver 14** and activate the HMI option in the Fbox.

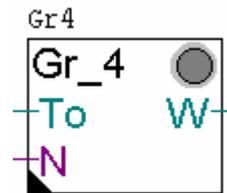
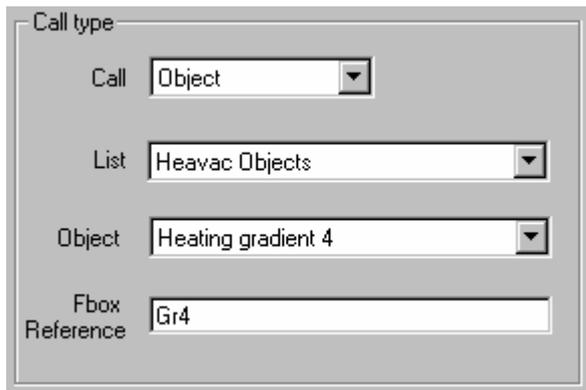


In the HMI-Editor, go to the [menu structure tab](#), select the menu item to call the object. In the [Call type](#) pane, select Call = Object. Choose the imported object list and the object of the corresponding Fbox. When you select a standard object, the Fbox reference field will appear. Type the Fbox name you have defined in the Fupla file.

Example:



Activate the HMI option in the Fbox.



Type a unique Fbox name in the Fbox properties and as reference in the HMI-Editor.

You can [re-import the objects](#) at any time to change the language or the texts length. The object names and the element texts will be automatically updated in each standard object.

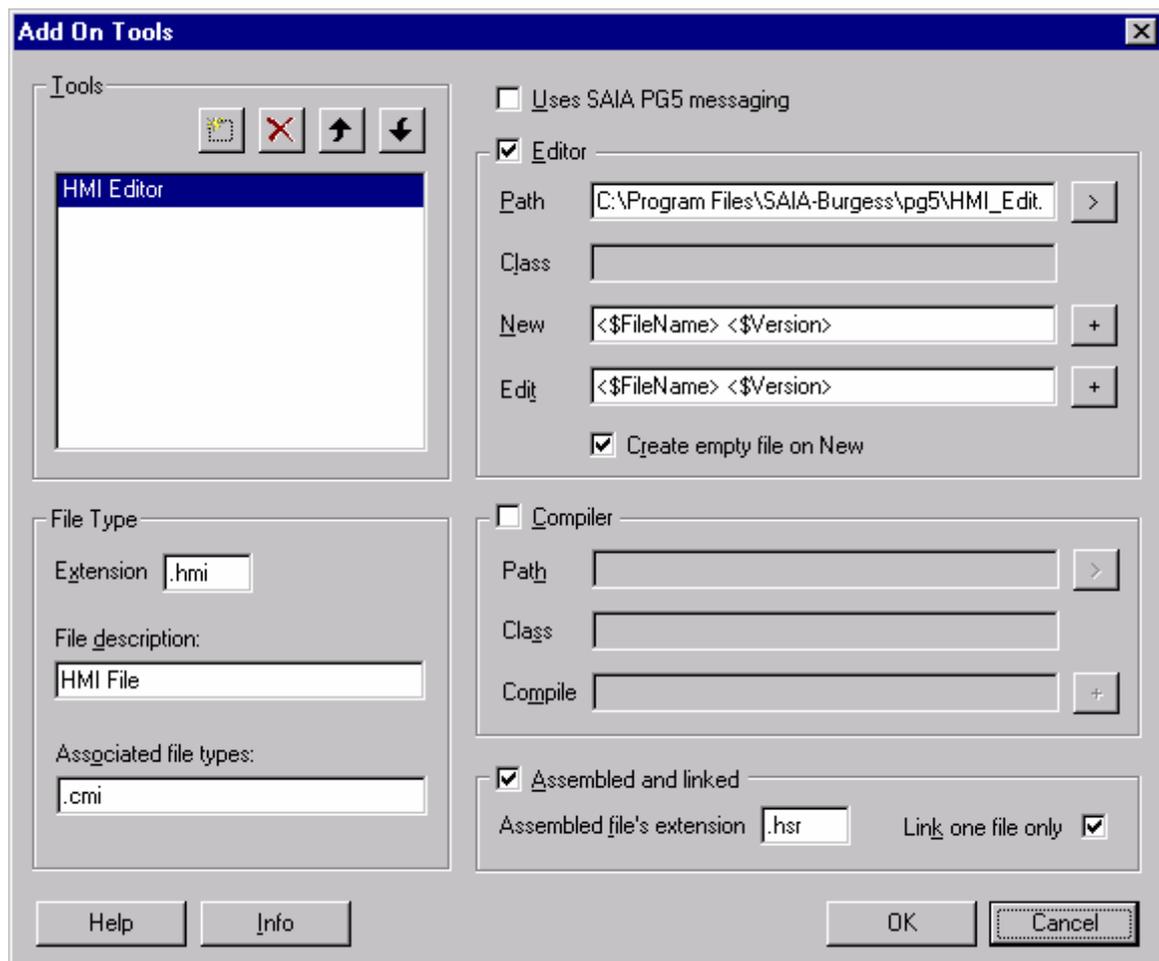
11 Installation and options

11.1 Add-On Tool installation in PG5

The HMI-Editor is automatically installed by installing PG5 programming software. In case of uninstal PG5 programming software, HMI-editor is automatically removed.

In case of installation error, or if the HMI Editor cannot be called, you can fix the problem by accessing the Add-On Tool feature in the Project Manager (menu Tool – Add-On Tool...).

In the Add-On Tool dialogue box, select the HMI-Editor and adjust the parameters according the following example. Check that the Path entry of the Editor points to the correct location of the HMI Editor.



The Info button can be used to view the Help file of the HMI-Editor.

11.2 Editor options

The editor options can be accessed by the Menu Option - Editor options.

The Copy tab allows you to choose the number of menu levels and if the objects are copied when copy-past is used in the Menu structure.



The Order tab allows you to define if the Texts and the resource lists must be sorted or not.

