EXOR
Tech-note

New Features in Designer Version 6.06

This technical note contains an overview of the new features included in Designer 6.06(00) and its companion firmware V5.61.

Contents

1 TrueType Font Support and Multilanguage 3
1.1 Configuring the project 3
2 Configuring Enhanced Alarms 5
2.1 Alarm Icons 6
2.2 Alarm Classes 7
2.3 Message Editor 8
2.3.1 Multilanguage Alarms 9
2.4 Previewing Alarm Messages 10
3 Macro Editor Programming Interface 11
3.1 Configure Macro Keys 13
3.2 Macro Editor 14

Tn245
Ver. 1.00
Copyright © 2006 Sitek S.p.A. – Verona, Italy
Subject to change without notice
The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.
www.exor-rd.com

1 TrueType Font Support and Multilanguage

Designer has been including a true multilanguage feature with full support of Windows Code Pages since version 6.03. This multilanguage facility handles different code pages for the different languages.
With Designer version 6.06 this feature has been extended to support TrueType Fonts (TTF) for all text information in the project, such as Static Text, Text Message and Alarm Messages.
This feature is requires hardware type FW60 so it is supported only by B-series panels. Firmware version 5.60 and above is required to support TTF text in the panel.
In Designer Panel Setup dialog, the new flag “TTF Text Support” is set to indicate the availability of this extended support for these objects.
The multilanguage feature can be used for a project by defining languages and setting appropriate code page for each language using the Configure Languages dialog. Designer also can use the TrueType Fonts available in Windows systems to provide different font faces associated to different scripts.
Setting the appropriate code page for each language you can have the correct character set (Cyrillic, Hebrew, etc.) available for different objects such as Touch cells, Static Text, Text Messages and Alarm Messages. Basically this Multilanguage feature with TTF support provides the ability to assign a unique font for each language to display strings as diverse as Chinese, Hebrew, Arabic and Central European.

1.1 Configuring the project

The Code Page (Script) selection can be done from the Configure Language dialog box (Project/ Languages… menu item). For each language you can define the associated script file (character set) and the font file that contains the selected script as shown in the figure below. In this way the languages required for the project are being defined globally.

Figure 1 – Configure Languages dialog box

The Layout panel of all Multilanguage supported objects such as Touch Cells, Static Text and Message field includes the Multilanguage radio button option (along with other font options Custom

and Fixed). Please, note that when the Multilanguage option has been selected the font dialog box does not give you the possibility to change the Font and the Script file selection because global parameters are used instead.

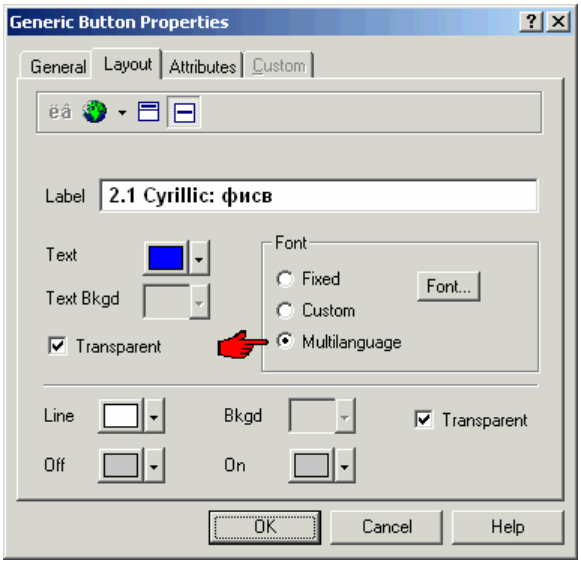

Figure 2 – Setting Font options in button properties

2 Configuring Enhanced Alarms

A new object oriented approach has been used for the design of alarms in Designer. With this new design, Designer provides a user interface as a tree object integrated in the project workspace to configure the alarm options. An example is shown in the figure below.

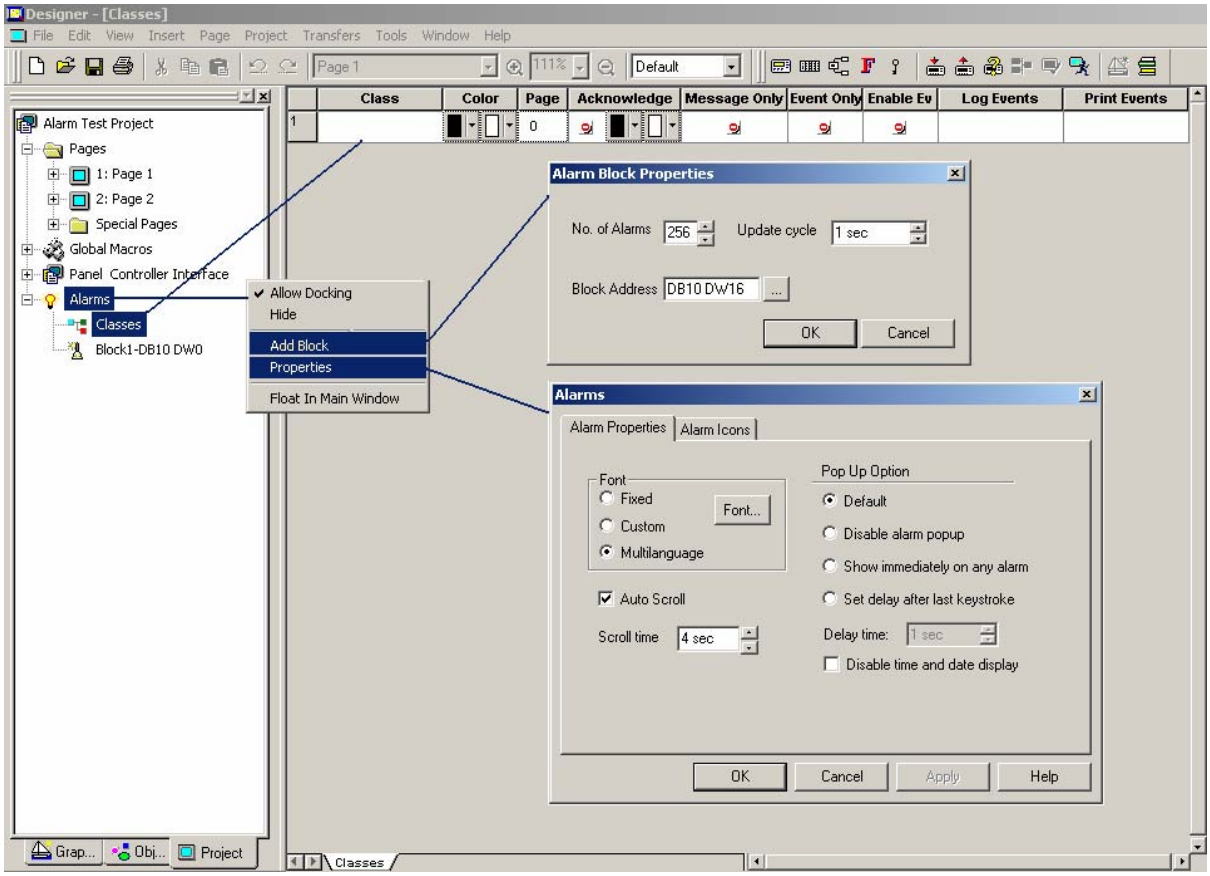

Figure 3 – The new alarm programming interface

With this new interface, the alarm message properties like display colors, page link, logging to event list, printing options etc. can be grouped as different **classes.** These group properties can then be assigned to individual alarms. The Alarm Pop-up options can be selected using the conventional property page accessible through the context sensitive menu command. The same menu also provides the command to add new alarm blocks.

With the introduction of the new **Multilanguage** TTF support, the alarm messages can be defined as true multilingual strings. Of course it is possible using different font settings as per the global language configuration of the project. Alarm messages can be configured to use multiple lines; the choice of different font sizes gives full control on this. The readability of the alarm messages in the screen of the panel must be kept in mind when reducing the font size and string length.
Designer 6.06 provides a preview window to present how the alarm will appear on the panel.
The extended font support is available to the Event List as well.

2.1 Alarm Icons

With the new TTF support, alarms and events benefit from a enhanced graphic support and a renovated programming interface.

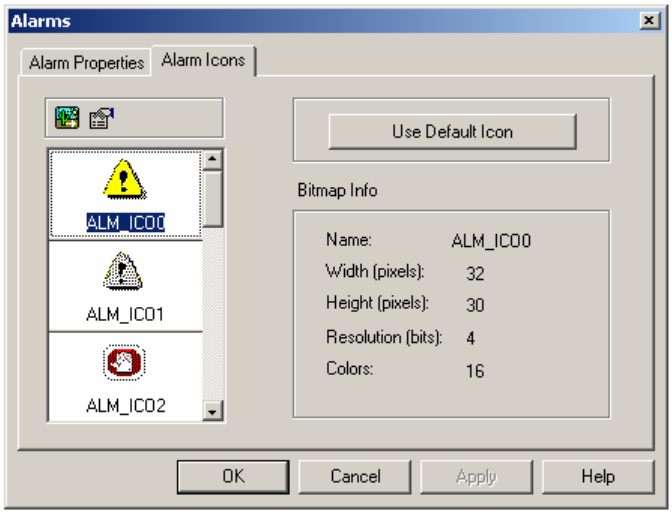

Figure 4 – Selecting Alarm icons

The Alarm Icons tab shows the predefined icons for the different alarm states. All the icons can be replaced or edited. “Use Default Icon” button is provided to reset all the images to default icons.

Enhances alarm presentation is possible only in products with TTF support, that is in the B-Series products.
For panels without TTF support, the icons are not supported and alarms will continue to have the existing way of indicating the states through alphanumeric messages. The table below shows a comparison between the alternative presentations.

Alarm State	Presentation with Fixed Font	Presentation in TTF Alarms
Alarm ON	A ON	
Alarm OFF	A OFF	
Acknowledge	ACK	
Event	E	
Event ON	E … ALARM ON	
Event OFF	E … ALARM OFF	
Event Acknowledge	E … ALARM ACK	
Page Link	*	

In TTF-enabled panels for the above symbols are used both for Alarm and Event pages

2.2 Alarm Classes

Alarm Classes provide a way of grouping the alarm message properties that can be applied to individual alarms. Different classes can be defined to group different set of properties. One of these classes must be selected while defining a new alarm message so that all the properties defined in that class will be applied to that alarm item.
For example the picture below shows the definition of three classes named as “Abnormal”, “Warning” and “Message”.

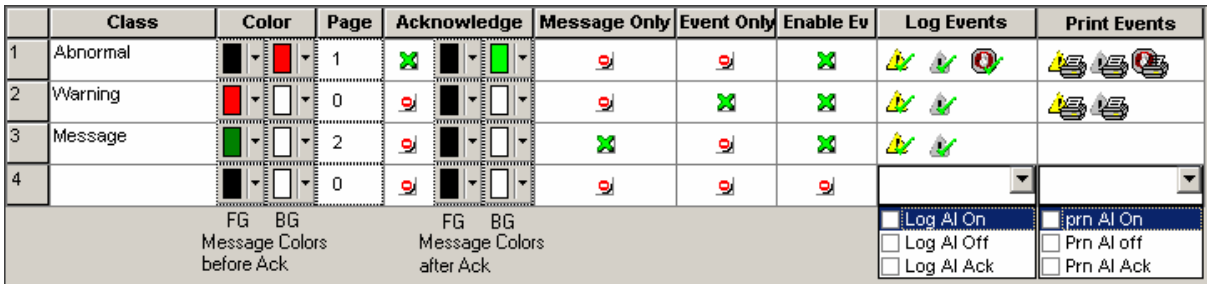

Figure 5 – Alarm classes

The figure above also shows the foreground (FG)/background (BG) colors applied before and after acknowledgement (if Ack option is selected) for any alarm message.
The Log Events and Print Events columns have a dropdown listbox which includes checkboxes with the available options for the class.

The table below shows the meaning of the different symbols.

Symbols	Meaning
	Option not selected
	Option selected
	Log Alarm On selected
	Log Alarm Off selected
	Log Alarm Ack selected
	Print Alarm On selected
	Print Alarm Off selected
	Print Alarm Ack selected

2.3 Message Editor

The block structure of alarms has not been changed. Alarms are still divided in 4 blocks. The alarm messages can be entered using the Alarm Message Editor window accessible by clicking on the desired Alarm Block node, under the Alarms in the workspace project tree.

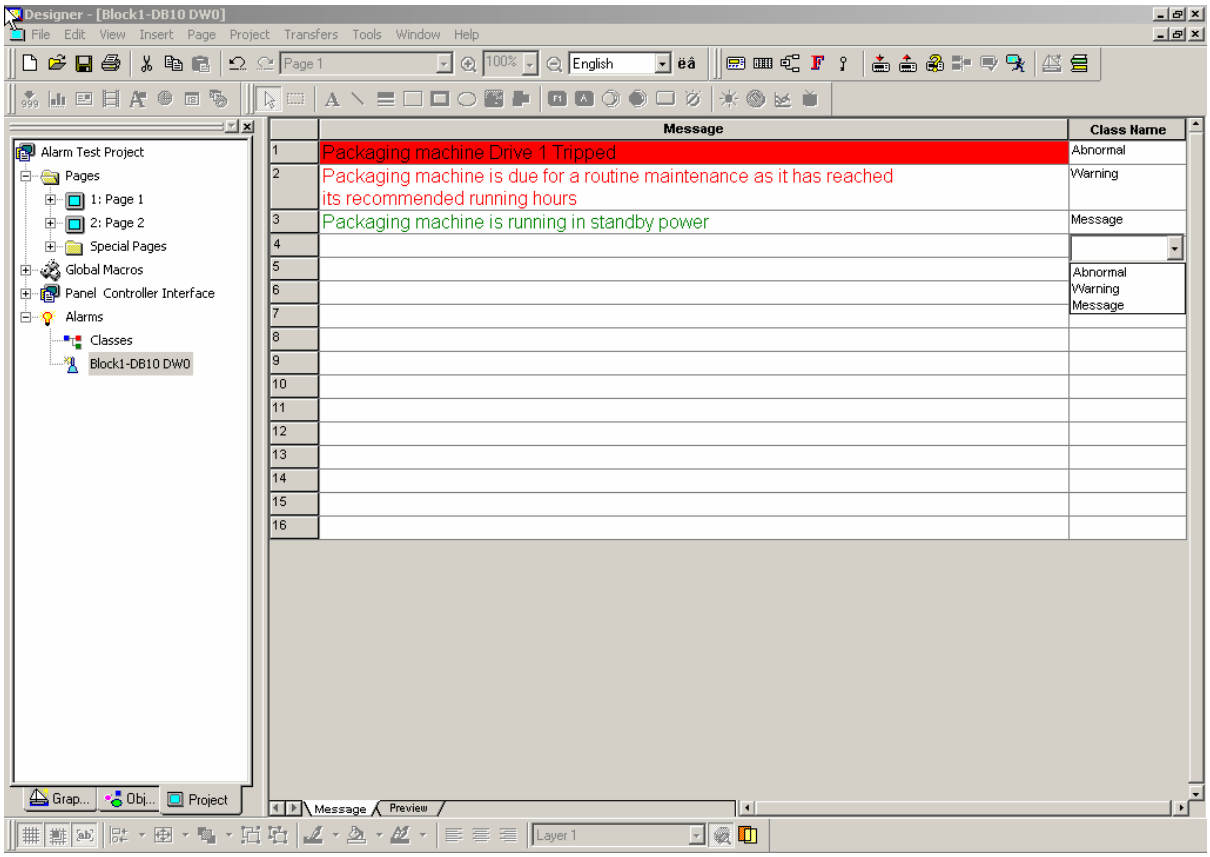

Figure 6 – Alarm Message Editor

2.3.1 Multilanguage Alarms

Alarm messages may be using different languages. To use the Multilanguage feature the project must have been configured already different languages set with different TTF font in **Multilanguage Configuration.**
For the TTF-enabled panels, by selecting the Multilanguage font option, the alarms can be defined in different languages set with different TTF fonts configured for the project. For example for a Chinese language/Simsun font, the message can be entered using Chinese keyboard as shown in the figure below.

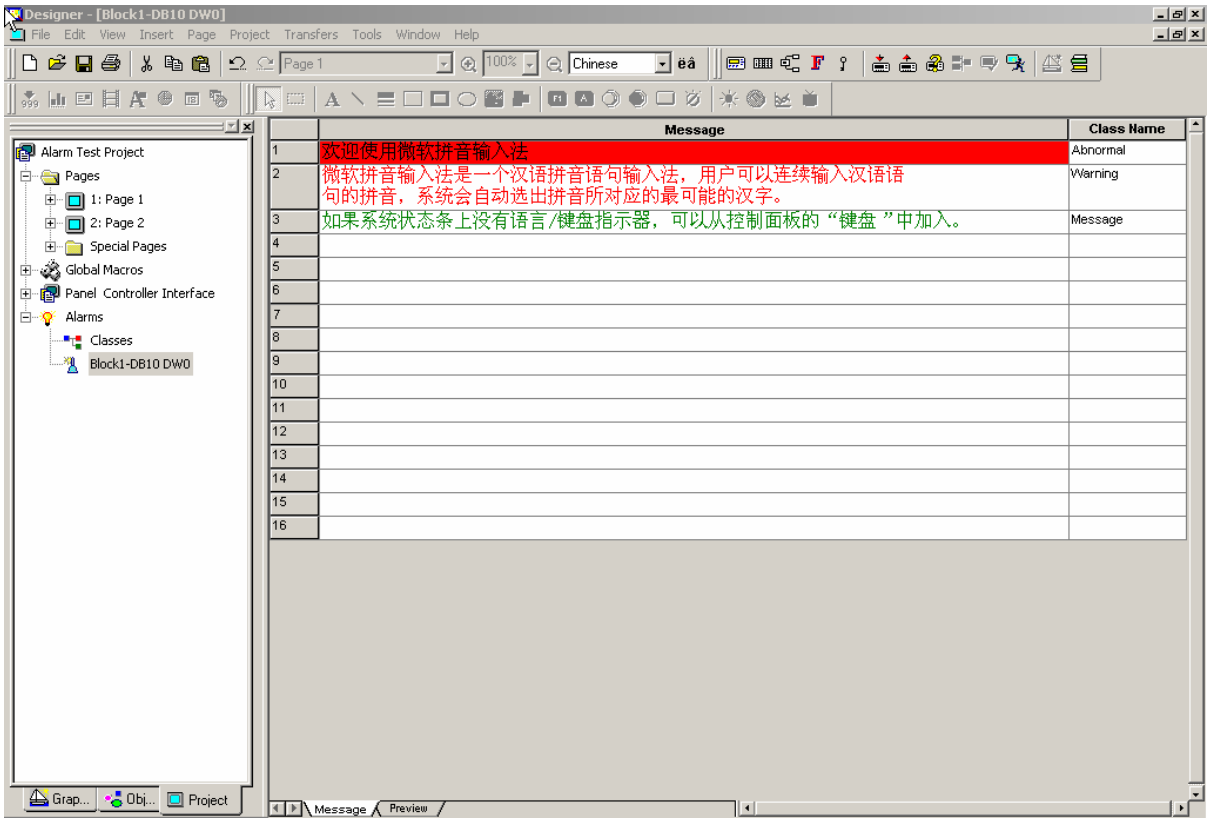

Figure 7 – Editing alarms in Chinese

2.4 Previewing Alarm Messages

Designer provides a preview window that will show how the alarm message will appear on the panel. Preview window is accessible by clicking on the “Preview” tab at the bottom of the message editor window. The figure below shows the preview window for the sample messages entered in the editor window shown above.

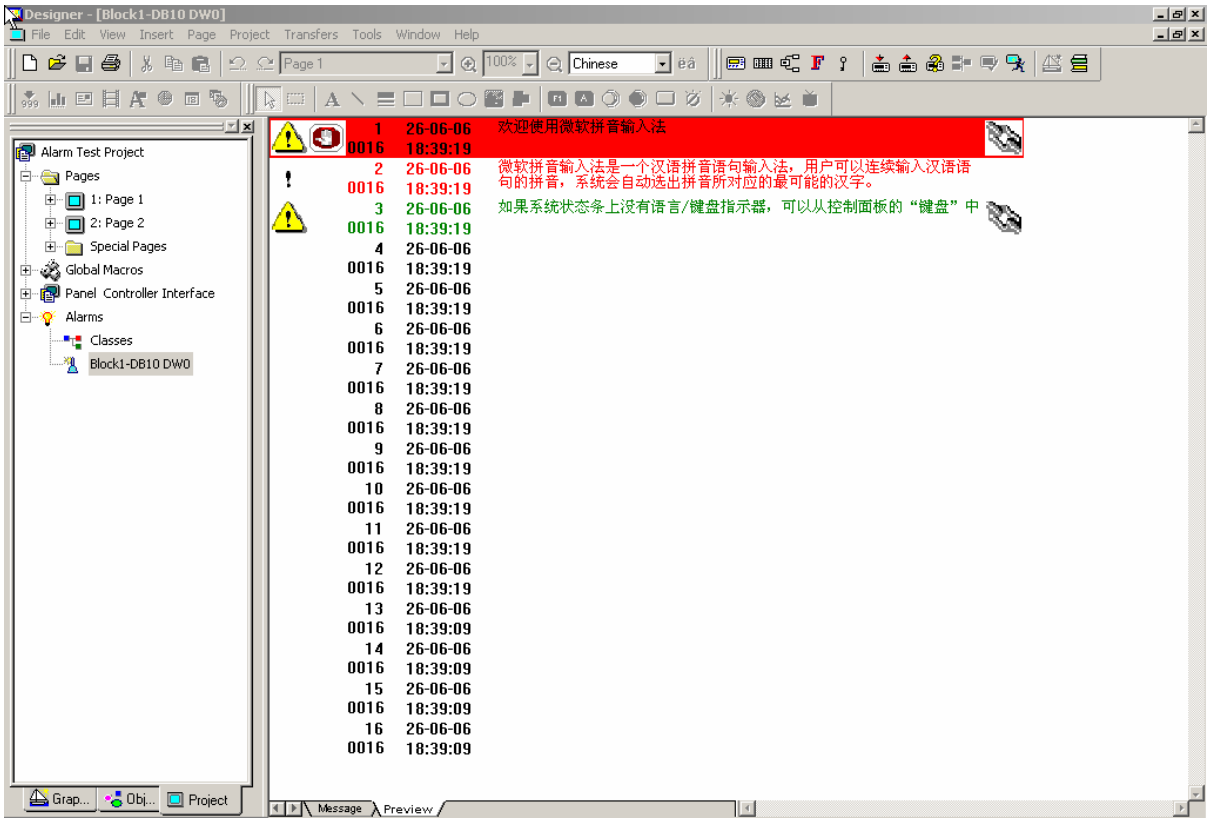

Figure 8 – Alarm preview window

3 Macro Editor Programming Interface

In Designer the macro programming interface is now integrated into the project tree in the workspace as shown in the figure below.

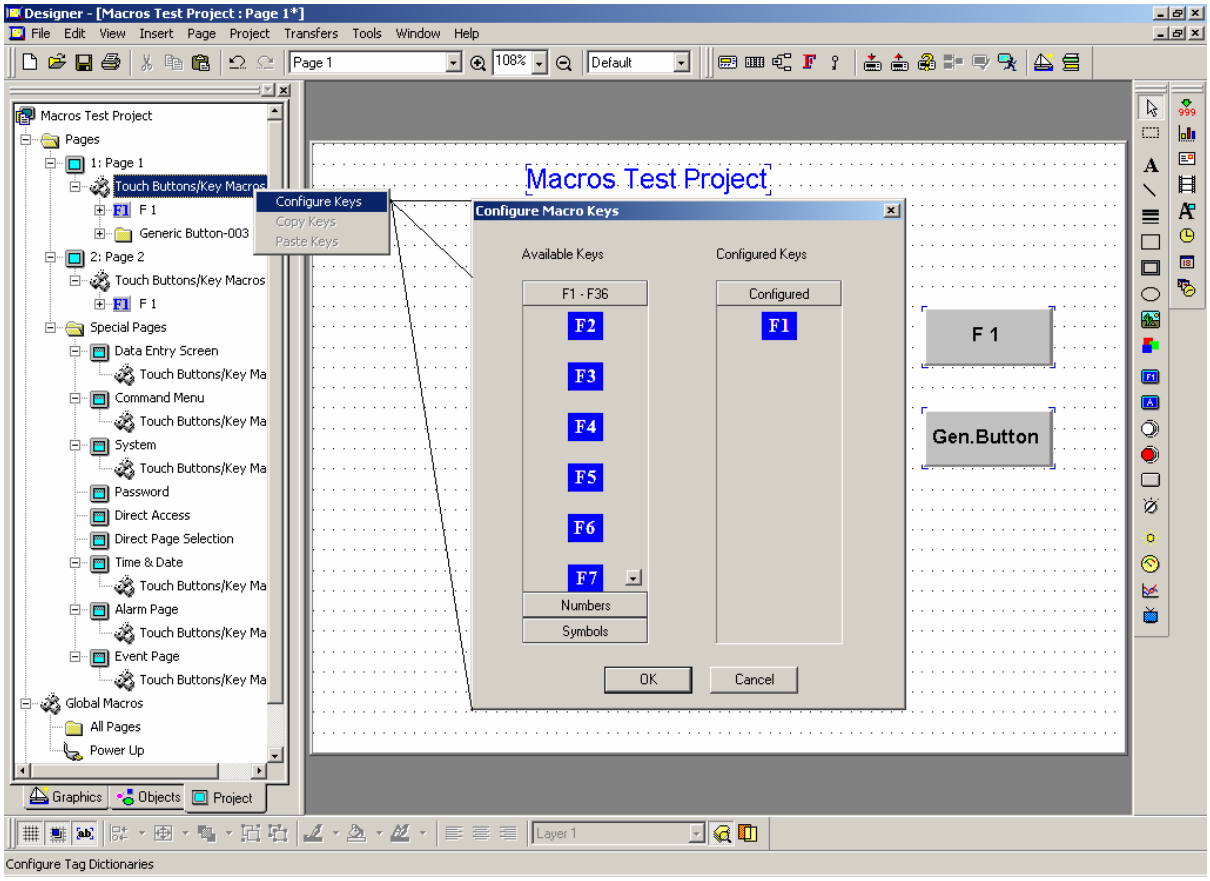

Figure 9 – The new Macro Editor programming interface

Function keys need to be configured under the required panel modes using the Configure Macro Keys dialog.
Generic macro buttons, instead, are added to the tree under the respective mode when the button is defined in the page. In the tree the button is identified with a generic string like “Generic button (ID: n) where in brackets it is shown the object ID as indicated in the Attribute tag of its Properties dialog box (see figure below).

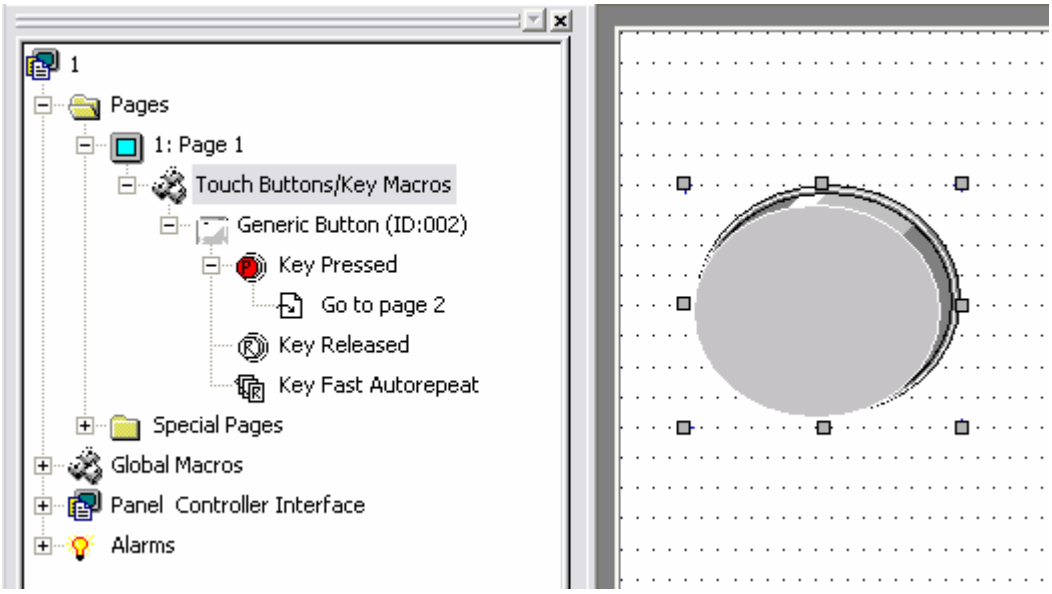

Figure 10 – Macro commands for Generic Buttons

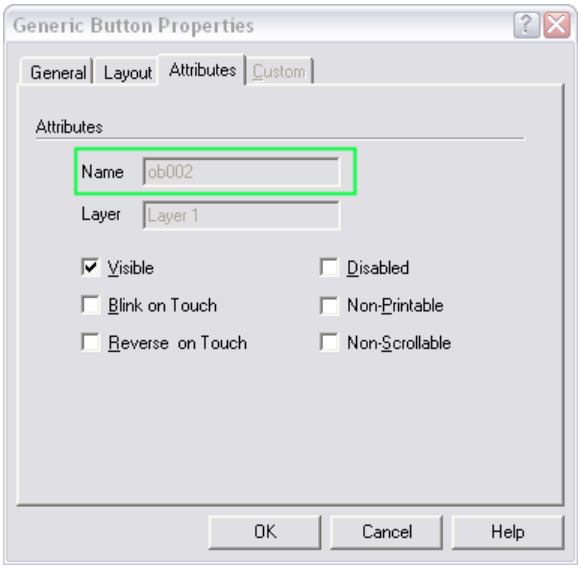

Figure 11 – Generic Button properties

The relation page objects and their tree view is the following:

- from the tree view a single click on the “Generic button” icon will highlight (select) the corresponding element in the project page.
- from a Generic button programmed in the page, the context menu provides the command “Locate Macro” that can be used to identify the corresponding macro folder in the tree view.

3.1 Configure Macro Keys

Keys required being functional for the project, need to be configured using **Configure Macro Keys** dialog shown in the figure below.

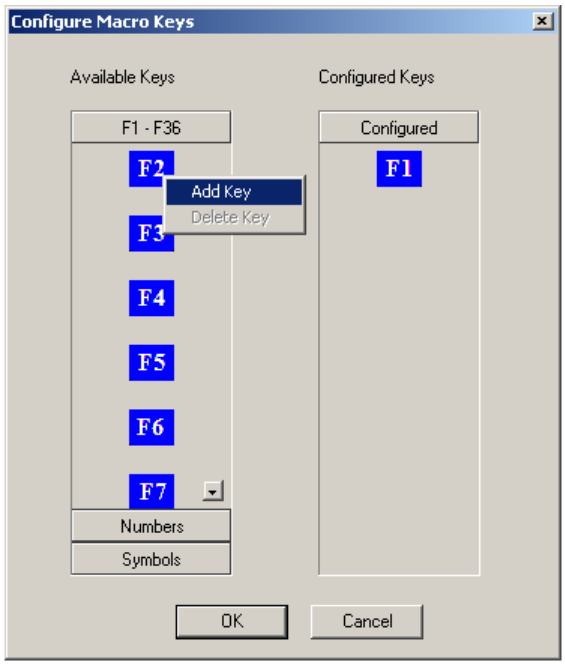

Figure 12 – The new Configure Macro Keys dialog box

This dialog can be opened through context sensitive menu (right mouse click) command “Configure Keys” on **Touch Buttons/Key Macros** node. The dialog includes, under Available Keys, different categories of function keys like F1 – F36, Numbers 0 – 9, and Symbols like “Help”, “Clear”, “Enter”, etc. Clicking on the appropriate category button will open the list of keys available under its category.

From the Available Keys list the key can be added to Configured Keys list with “Add Key” context menu command. More easily a double mouse click on any key from the Available Keys list will add that key to Configured Keys list. Also this context menu includes a “Delete Key” command which can be used to delete any key from Configured Keys list. When the dialog is closed the keys configured here will be added to the project tree under the respective **Touch Buttons/Key Macros** node.

3.2 Macro Editor

The Macro Editor lets you to program/assign the macro commands to the panel keys configured under different modes and to the generic buttons defined on any page to perform application specific functions. The figure below shows the **Edit Macro List** dialog which can be accessed through the context sensitive menu command “Edit Macro List” on any key state under the panel key tree of the workspace.

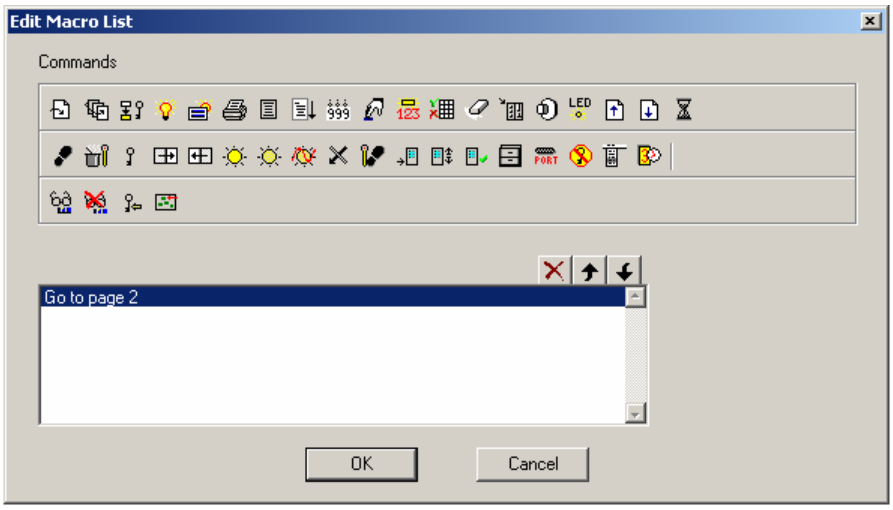

Figure 13 – Macro editor window

The Generic macro button properties page, also includes a button called “Macro…” that can be used to access the Macro List as shown below.

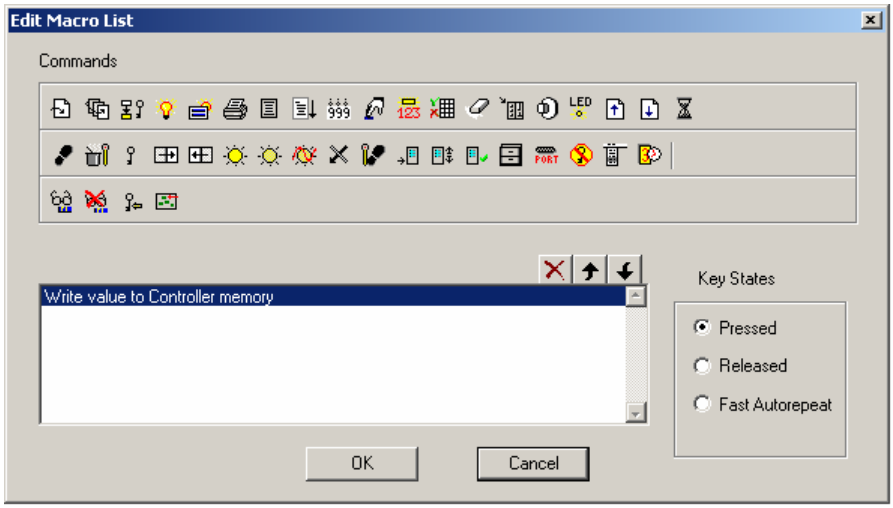

Figure 14 – Macro editor window