DIMETRONIC
signals

# ID2303 PCGE

**DOCUMENT** : PCGE User Manual

**SUMMARY** : This document is a PCGE user manual

**REFERENCE** : ID2303 / M0001

**ISSUE** : 1.1

**STATUS** : Provisional

**DATE** : 15/04/02

**LOCATION** : \\SERVER_I&D\ ID2303 \FORMAL\Documentacion\PCGE Suite Tool\General.\M0001_v

**FILE** : M0001.DOC

| | AUTHOR | CHECKED BY | APPROVED BY |
|---|---|---|---|
| NAME | Mónica Muñoz | Santiago Fernández | Celestino Martínez |
| SIGNATURE | | | |
| DATE | | | |

# Index

## Document History

| Issue | Date | Change | Author |
|---|---|---|---|
| 0.1 | 29/6/01 | Creation | Mónica Muñoz |
| 1.0 | 04/07/01 | After revision ID2303/0048 | Mónica Muñoz |
| 1.1 | 15/04/01 | - New parameter of fill rectangle entity, *Display Fill*.<br>- The new version had been developed in Windows 2000.<br>- New feature; import GPC files version format 1.07 and 1.08<br>- New tool: InsertPrefix2GPCMnes | Mónica Muñoz |
| | | | |
| | | | |
| | | | |
| | | | |

## Abbreviations

| | |
|---|---|
| **CAD** | **C**omputer **A**ided **D**esign |
| **CLT** | **C**lassic **L**ogic **T**able |
| **DtMC** | **D**irect **t**o **M**nemonic **C**onvert |
| **ELT** | **E**xtended **L**ogic **T**able |
| **GPC** | (**G**raphics **PC**). Text file graphic format. |
| **HGRA** | **H**erramienta **Grá**fica para HP WorkStation — *former tool used for graphic configuration. This tool was an ancestor of PCGE Tool.* |
| **HPAN** | **H**erramienta **Pan**talla para HP WorkStation — *former tool used for graphic configuration. This tool was an ancestor of PCGE Tool.* |
| **LT** | **L**ogic **T**able |
| **MAA** | **M**nemonic **A**ssignment **A**ssistant |
| **MGR** | **M**nemonic **Gr**aphic file |
| **PCGE** | **PC G**raphic **E**ditor |
| **SCC** | **S**mall **C**ontrol **C**enter |
| **TBL** | **T**a**BL**e |

## References

| | | |
|---|---|---|
| [1] | SV-01<br>Edición 1 – Junio 93 | Norma de Sistemas Vidiogáficos para Enclavamientos y Telemandos (RENFE) |
| [2] | | |
| [3] | | |
| [4] | | |
| [5] | | |
| [6] | | |
| [7] | | |
| [8] | | |
| [9] | | |
| [10] | | |
| [11] | | |

# 1 Introduction

## 1.1 Conventions Users in this Manual

**Figures.-**

i.e.

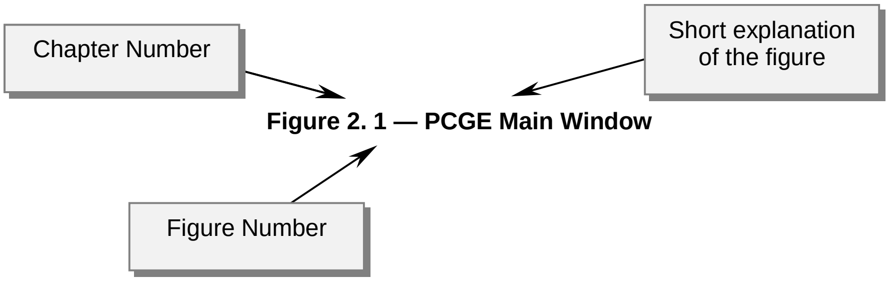
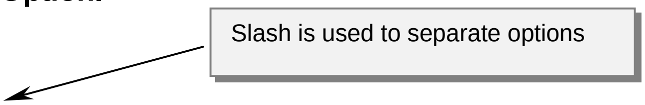


**Figure 2. 1 — PCGE Main Window**

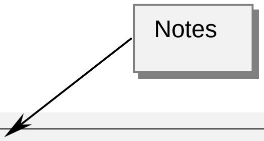
**Menu Option.-**

i.e.



*Show/Other Toolbar*

**Notes.-**

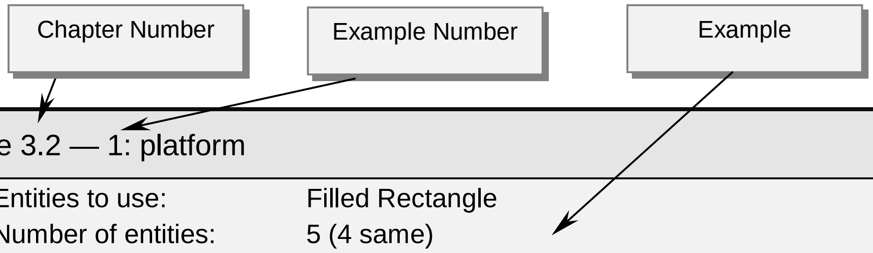


***Note***

So far, buttons are only used in DIMETRONIC graphics because DGRA Options match with NOI options generated by DIMETRONIC Command Tool HORD.

**Example.-**

i.e.



Example 3.2 — 1: platform

Entities to use: Filled Rectangle
Number of entities: 5 (4 same)
Process:

- Click on button *Filled Rect* in Block Toolbar. ▣
- Move mouse pointer to the block edition view to start drawing the rectangle.
  Note that mouse pointer has a cross aspect. +
  Click on desired point and drag the mouse to outline a rectangular shape. Release left button when rectangle size is OK.

## 1.2 Document Overview

This document contains 6 chapters:

Chapter 1 introduces the PCGE User Manual.

Chapter 2 describes the PCGE User Interfaces.

Chapter 3 describes how to use PCGE.

Chapter 4 describes how to install and set up PCGE on a PC.

Chapter 5 describes the compatibility of different versions.

Chapter 6 describes other tools related to PCGE.

## 1.3 Purpose of the Manual

This manual is aimed to design engineers with responsibility for designing GPC graphics. User knowledge of Windows is assumed, and also a basic knowledge of CAD tools.

Furthermore, a basic knowledge of signalling will be valuable to understand some of the concepts of this manual.

When finishing this manual, a person will have acquired advanced knowledge to use PCGE.

## 1.4 Definitions

Terms used in this manual:

**Entity** Single graphic indivisible elements. Each entity is the minimum graphic information capable of being represented. Entity samples are a line or a circle.

**Block** Entity groups.

**Insertion** Block instances. Each insertion represents all block entities in the design, in each block instance.

**Label** Some text needs to be in the design, but not grouped into blocks. This can be accomplished with labels.

**Logic Table** In a set of bits, a Logic Table establishes links between bit values and entity colours for every combination.

**Behaviour** Logic Tables groups with significance inside blocks. Behaviours are inherited by insertions.

**Design** Two-dimensional space where labels and insertions are contained. It is the visible part of an installation.

## 1.5 PCGE ToolSuite Overview

PCGE Tool Suite version 3.1 comprise the following tools:

**PCGE.-** The graphic editor in itself, explained in this manual.

**DtMC.- D**irect **T**o **M**nemonic **C**onverter. This tool performs conversion from direct bits into mnemonics in two steps. First step consists in changing all direct bits in one design into unassigned mnemonics. Second step consists in automatically assign a mnemonic from a mnemonic's list supplied from an external list. The automatic assignation algorithm is based in Line Housing Card and Bit mapping of a SCN41 card.

**GPCLibraryManager.-** This tool manages block elements between two different GPC files. Its main goal is cost saving when creating new designs because blocks from another design can be imported safety.

**GPCCompare.-** Is a powerful tool able to compare two GPC using GPC format, not a line-to-line comparation. It is a useful tool for control management of versions.

The contents of this manual are compatible with the next product versions:

- PCGE v. 3.1
- GPCCompare v. 1.0
- DtMC v. 1.0.0.1
- GPCLibraryManager v. 4.0

# 2 PCGE User Interface

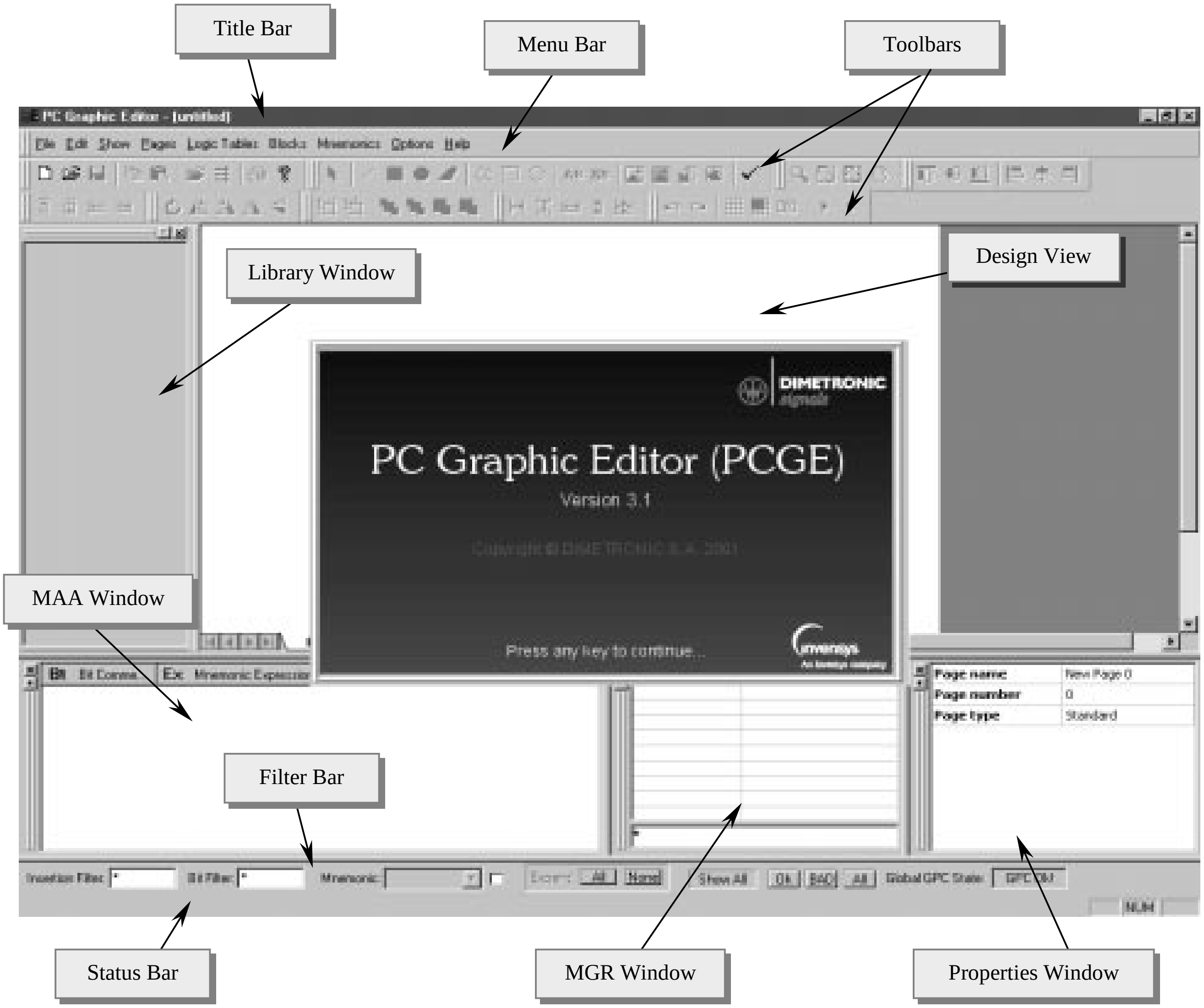
PCGE is a graphic tool with a typical Windows interface. A splash window is showed during the opening session of the application.

The user may interact with the graphic tool using keyboard and mouse. The user may view the results from graphic displays and by printing the display.

## 2.1 General Description

The main PCGE Window consists of six elements. Five of them are views and the last one is a set of toolbars.

In the figure 2.1 the user can show the initial aspect of PCGE.



| Page name | New Page 0 |
|---|---|
| Page number | 0 |
| Page type | Standard |

**Figure 2. 1 — PCGE Main Window**

The top of the screen contains a title bar with the main control menu of the all standard windows and a menu bar for controlling the graphic editor. When using this menu the user may select all operations provided by the PCGE.

The general aspect of this main window consist on five views:

- one main view, which holds all design and library views, used by the designer, separated by tabs.
- one view to show block palette used for all blocks existing at this moment in the file.
- one MAA view to show all insertions that contain bit comments programmed with mnemonics.
- one view to show properties of the selected object.
- one view to show MGR mnemonics used for MAA view. The list of valid mnemonics is taken from an external file.

There are several toolbars below the main menu, which make the interaction between the user and PCGE easier.

### 2.1.1 Design Page Window

In this page the designer may create, edit and design a track layout. There are two possible types of views inside this area: the design view and block edition view.

When a design view is the active view, the designer can insert blocks in the respective page for creating or updating a layout diagram. If the active view is block edition view the user can create and edit the block that will be inserted in a design view.

All these pages will be separated by tabs, which include the page's name. If the user wants to change from current page to another, he must click on the name of the page desired in its respective tab. The selected page then becomes the page displayed. Tabs are placed at the bottom of the Design Page Window. The user can close the block edition views.

Figure 2.2 shows a block edition view with several entities.

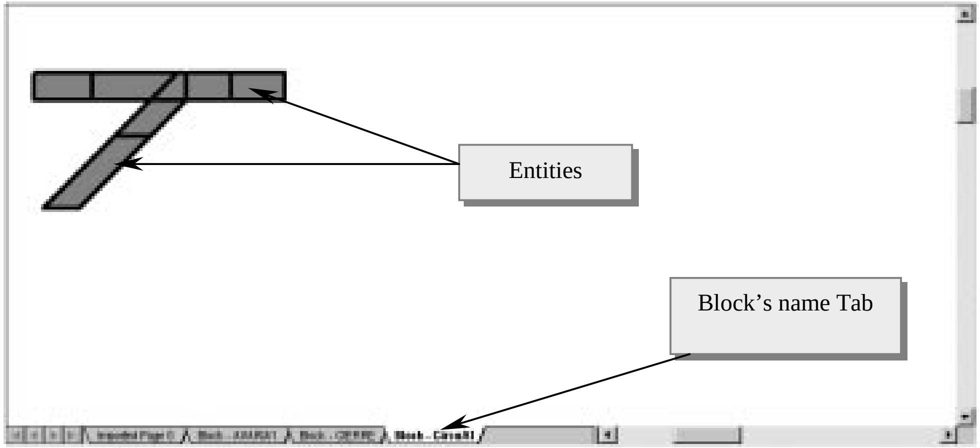


**Figure 2. 2 — Block Edition View**

Figure 2.3 shows a track diagram in a design view with several elements.

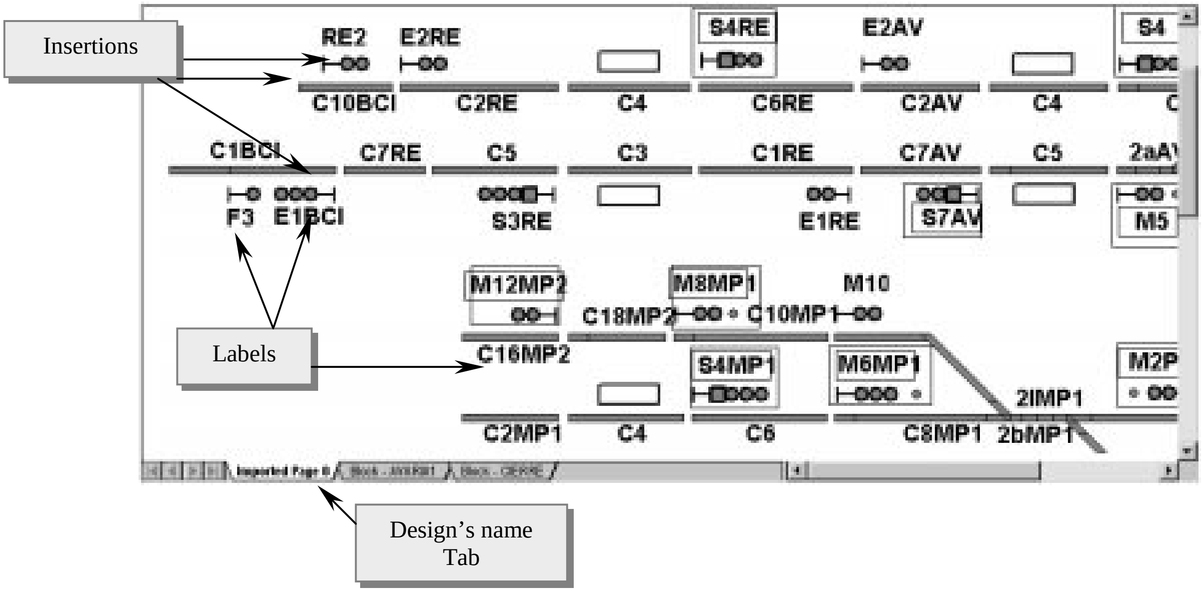


**Figure 2. 3 — Design View**

### 2.1.2 Library Window

In this window are shown all blocks created by the designer. They are ready to be placed onto de design view.

All blocks are alphabetically shorted so that is easy to find them.

Blocks can be of two types: button and not button. A button block name starts with the special character '>'. The tool automatically sets this character. They can be found at the end of the Library Window.

Figure 2.4 shows this window.

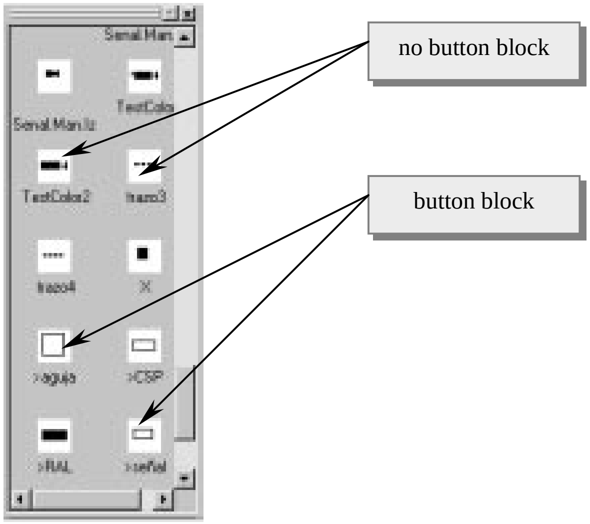


**Figure 2. 4 — Library View**

### 2.1.3 Library Detail Window

This window allows user to see a detailed view of the selected block in Library View. It is a splitter window so it can be set to the desired size. If user moves across the Library Window with the cursors the image of selected block is shown in detail in the detailed window also.

If this window is not bet visible, place cursors in the bottom border of the library view, just below the horizontal scroll. When cursor changes its style, then click and drag up (see figure 2.5).

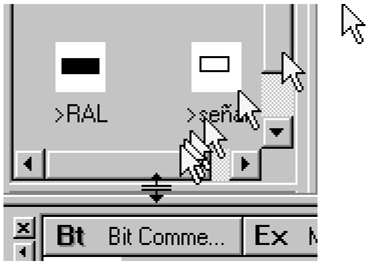


**Figure 2. 5 — How to show the Library Detail Window**

Figure 2.5 shows how to enlarge this window if is initially too small.

Once done this operation, user is ready to get a detailed view of selected block.

Figure 2.6 shows the aspect of a block in the detail window:

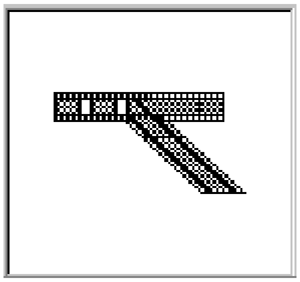
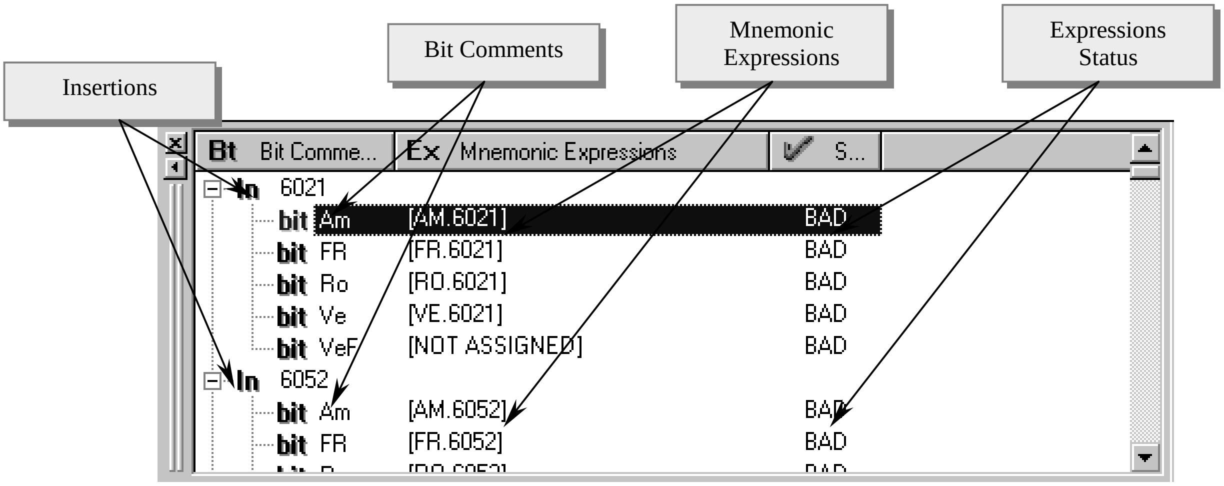


**Figure 2. 6 — Library Detail Window**

### 2.1.4 MAA Window

This window shows all insertions containing bit comment of type mnemonic. A tree control displays the insertions with all of its mnemonic bit comments, the boolean equation entered for each bit comment and its state (OK or BAD)

Figure 2.7 shows an example.

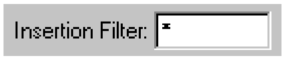


```
⊟ In  6021
   bit Am    [AM.6021]          BAD
   bit FR    [FR.6021]          BAD
   bit Ro    [RO.6021]          BAD
   bit Ve    [VE.6021]          BAD
   bit VeF   [NOT ASSIGNED]     BAD
⊟ In  6052
   bit Am    [AM.6052]          BAD
   bit FR    [FR.6052]          BAD
```

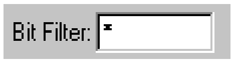
**Figure 2. 7 — MAA View**

Initially all equations are shown expanded, but clicking over the ⊟ symbol in the left of the insertion name the particular insertion’s set of equations are collapsed. Once collapsed, the symbol ⊞ can be used to expand again the set of equations.

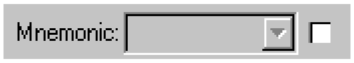
Clicking the button Expand All/None in the bar below this window can also do this operation. Buttons to filter MAA information as mnemonics or bit comments are also placed on this bar.



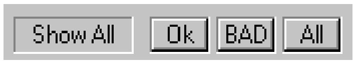
This is an edit box to filter insertions in MAA view by their name. This control allows wildcards. For example, next filters are valid values: X* (insertions whose name begins with ‘X’), *2 or 60*2.



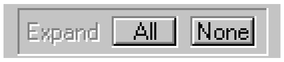
This edit box is used to filter insertions by their bit comment name. It works in a very similar way as the previous one.



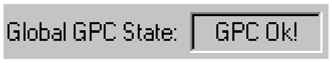
Another way to filter information is by mnemonics used. Mnemonics are used as operands in boolean expressions. In order to perform this filter, a MGR file must be loaded. MGR files are external ASCII files containing mnemonics in a single column. Mnemonics normally comes from Westrace or SCN41. Click the checkbox beside the combo box to set this control. When a mnemonic is entered, all expressions containing this mnemonic as operand are filtered.



Those buttons are used to expand/collapse information in MAA window.



Those buttons filter boolean equations according to their status. A boolean equation is marked as OK if all of its operands are mnemonic recognised in the MGR file and it is syntactically correct. Conversely, it is marked as BAD if it is syntactically incorrect or if one of their operands is a mnemonic not found in the MGR file.



Indicator of GPC status.

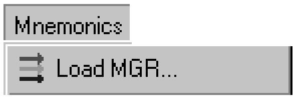
### 2.1.5 MGR Window

This window holds the valid mnemonic list to be used in programming mnemonic bit comments. Searching of mnemonics are simplified using a search pattern. Wildcards are allowed.

If GPC file contains insertions with bit comment of mnemonic type, an external MGR is required.

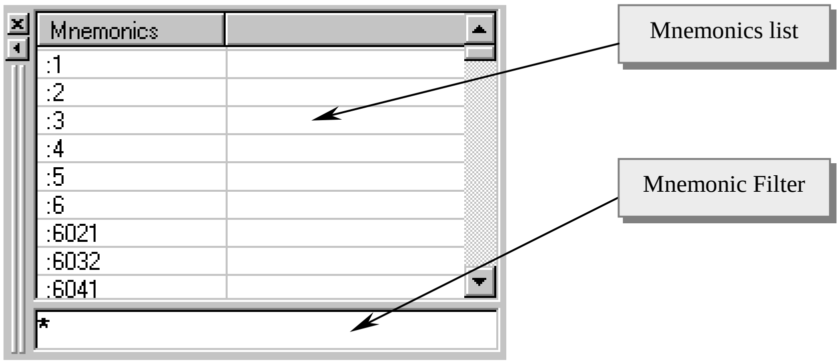
There are two methods to load a MGR file:

1. By means of menu option *Mnemonics/Load MGR...*



2. By means of toolbar button '*Open a MGR file*' from default toolbar.

Figure 2.8 shows all different parts of MGR window:



:1
:2
:3
:4
:5
:6
:6021
:6032
:6041

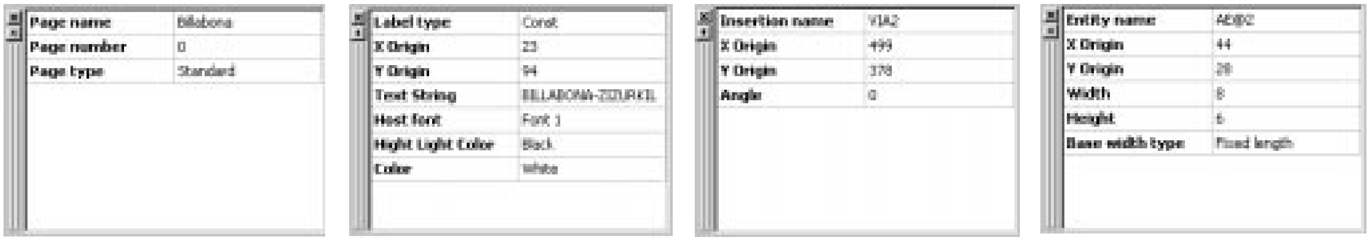
*

**Figure 2. 8 — MGR Window**

### 2.1.6 Properties Window

This window displays all the properties of the different objects selected. Depending on the selected element, the properties in this window can vary. Editable properties can be edited from this window.

Figure 2.9 shows different aspects of this window.



| | |
|---|---|
| Page name | Billabona |
| Page number | 0 |
| Page type | Standard |

| | |
|---|---|
| Label type | Const |
| X Origin | 23 |
| Y Origin | 94 |
| Text String | BILLABONA-ZIZURKIL |
| Host font | Font 1 |
| Hight Light Color | Black |
| Color | White |

| | |
|---|---|
| Insertion name | VIA2 |
| X Origin | 499 |
| Y Origin | 378 |
| Angle | 0 |

| | |
|---|---|
| Entity name | AD02 |
| X Origin | 44 |
| Y Origin | 28 |
| Width | 8 |
| Height | 6 |
| Base width type | Fixed length |

**Figure 2. 9 — Properties Windows: Page, Label, Insertion and entity (Filled Rectangle)**

Fields of Properties Window can be edit box or combo box. In order to edit an edit box just click on the edit box and change the value. In case of a combo box, click on the field and click again to show the unfold button.

### 2.1.7 Toolbars

PCGE has eleven toolbars including menu bar and status bar.

**Default**

This toolbar group all operations related with files (opening, writing, printing, etc).

Figure 2.10 shows all buttons with a brief description.

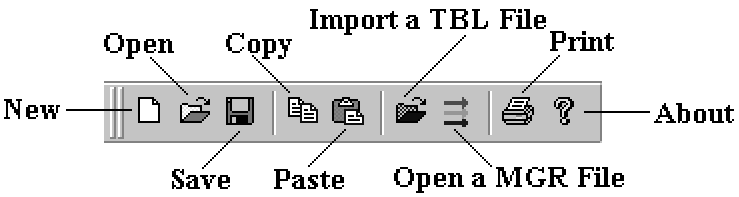


**Figure 2. 10 — Default Toolbar**

Menu option *Show/Toolbar* allows showing or hiding this toolbar. Check the symbol ✓ in the left to know if toolbar is active.

**Library**

This toolbar groups all the functionality to create and edit blocks. It has all the entities used by PCGE.

Figure 2.11 shows a brief description of all the buttons in Library Toolbar.

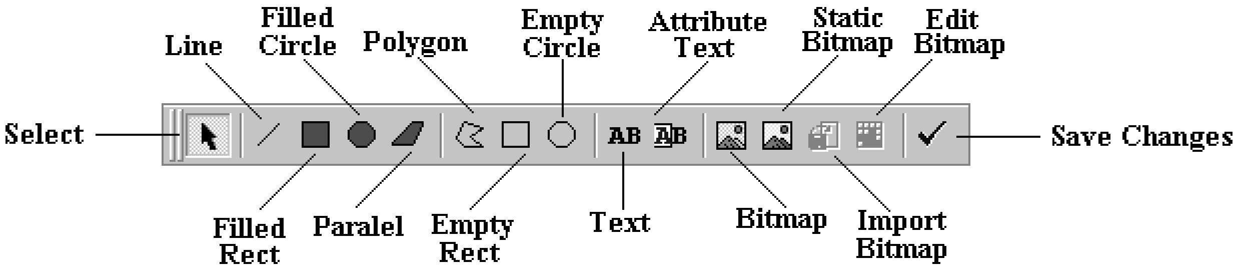


**Figure 2. 11 — Library Toolbar**

Use menu option *Show/Other Toolbar* to show or hide it. Check or uncheck Library in the subsequent dialog.

### Zoom

It groups zoom functionality. Figure 2.12 briefly describes Zoom toolbar.

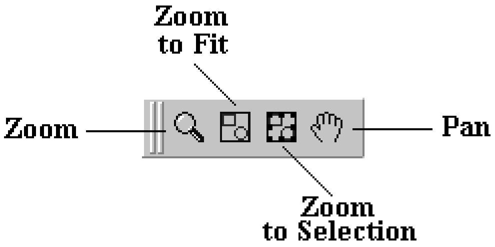


**Figure 2. 12 — Zoom Toolbar**

When clicking the first button of the toolbar (Zoom mode) the zoom mode enters. When in zoom mode, clicking left and right mouse button simultaneously will produce in a zoom factor. Once obtained the desired factor click on *Select* button of *Library* toolbar to return to selection mode.

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Zoom option in the subsequent dialog.

### Align

It groups align options for selected elements (insertions, labels or entities). Note that align can be separated in two groups, vertical and horizontal alignment.

Align takes effect when two or more elements are selected. Last element selected will be taken as reference.

Figure 2.13 shows Align toolbar.

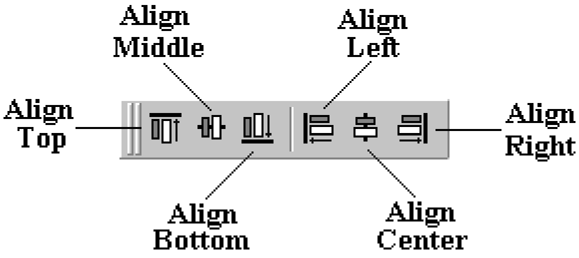


**Figure 2. 13 — Align Toolbar**

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Align option in the subsequent dialog.

### Nudge

This bar allows a fine movement adjusted to the grid snap of the selected element. Cursor keys can also perform those buttons functionality.

Figure 2.14 shows Nudge toolbar.

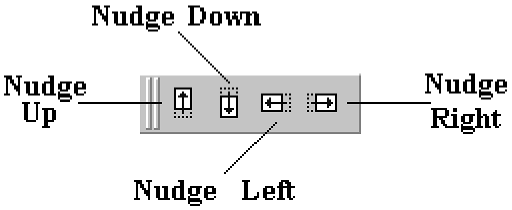


**Figure 2. 14 — Nudge Toolbar**

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Nudge option in the subsequent dialog.

### Rotate

This toolbar groups options rotate insertions. Rotation angles allowed are 90, 180 and 270 degree.

The first button yields a dialog from which user can select a rotation value to rotate selected insertions. The other two buttons apply successive 90 degrees rotations in the sense marked by the arrow in each stroke. Mirror buttons are deactivated in PCGE 3.1 version.

Figure 2.15 shows Rotate toolbar.

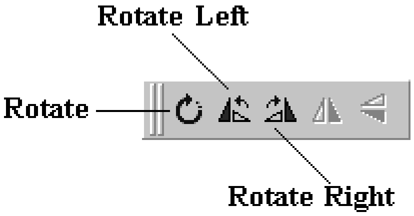


**Figure 2. 15 — Rotate Toolbar**

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Rotate option in the subsequent dialog.

### Structure

This toolbar groups relative position between elements.

The first two buttons brought selected element to the front or send it to back. The other buttons do the same but only one level each stroke.

Figure 2.16 shows Structure toolbar.

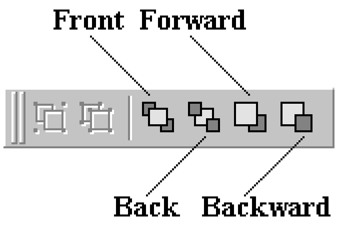


**Figure 2. 16 — Structure Toolbar**

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Structure option in the subsequent dialog.

### Layout

This toolbar groups options that allow user to re-distribute the elements relative position.

Two first buttons are activated in design view, and they distribute space between selected insertions.

The rest of buttons are active when in block view, and they let to use graphical properties from one object to be copied in other.

Figure 2.17 depicts the Layout toolbar:

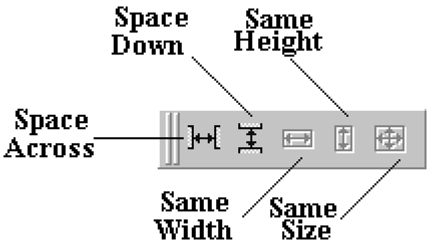


**Figure 2. 17 — Layout Toolbar**

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Layout option in the subsequent dialog.

### Canvas

This toolbar groups options related to grid and as well as undo/redo.

Figure 2.18 shows Canvas Toolbar.

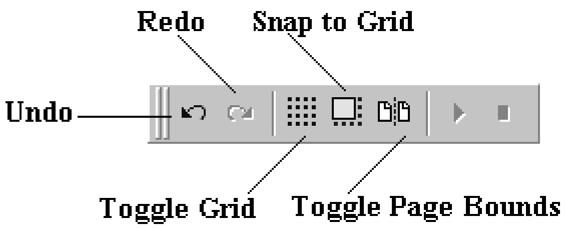


**Figure 2. 18 — Canvas Toolbar**

First group are undo/redo operations. Undo/redo functionality is standard.

Second group controls grid. First button show/hide grid. Second is **Snap to grid** according to the snap selected. If grid displaying has the same value than snap, then the movement is forced to snap to the same points than grid.

Use menu option *Show/Other toolbar* to show/hide this toolbar. Check/uncheck Canvas option in the subsequent dialog.

**Status Bar**

This toolbar shows general PCGE information.

Mouse co-ordinates are shown at the left side so that user can notice the position of mouse pointer in edition of design and block at every time. Current view zoom factor is shown beside mouse co-ordinates.

There are three spaces at the right side of the status bar. Those spaces are reserved for flags. When empty the operation is disabled, in case else is enabled.

First flag is CAPS LOCK, second is NUM BLOQ and third is SCROLL LOCK.

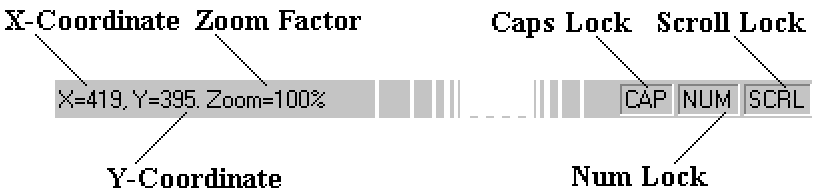

X=419, Y=395. Zoom=100%

**Figure 2. 19 — Status Bar**

Use menu option *Show/Status* Bar to show/hide this toolbar.

# 3 Using PCGE

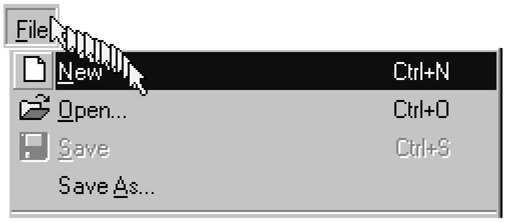
In this section user will be explained how to use PCGE. When reading this section user will be able to create single-track diagrams.

## 3.1 Creating and Open a design

### 3.1.1 Creating a new design

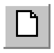
The first step is creating a new design. This operation can be done in two different ways:

1. Using menu option *File/New*.



The picture shows that a shortcut Ctrl+N is also associated to this action.

2. By means of toolbar button *New* in *default toolbar.*



When a new design is created, a set of default characteristics is attached to the new design:

- One page with resolution of 640 x 480.
- The file is set to be compressed when saved.
- No insertions, no blocks and no labels.
- Type and page description are shown in the properties view.
- The set of colours are:
  - black(0)
  - white(1)
  - flashing white(2)
  - red(3)
  - flashing red(4)
  - yellow(5)
  - flashing yellow(6)
  - green(7)
  - flashing green(8)
  - blue(9)
  - flashing blue(10)
  - light blue(11)
  - flashing light blue(12)
  - grey(13)
  - flashing grey(14)
  - pink(15)
  - flashing pink(16)
  - violet(17)
  - flashing violet(18)
  - brown(19)
  - acoustic signal(20)
  - transparent(21)
  - alarm(22)
  - flashing brown(23)
- The set of fonts are:

- Default font (0) System bold, 12
- Font 1 (1) Arial, 14
- Font 2 (2) Arial, 28
- Font 3 (3) Arial bold, 10
- Font 4 (4) Arial, 2
- Font 5 (5)Arial, 21
- Font 6 (6) Arial, 18
- Font 7 (7) Arial bold,24
- Font 8 (8) Arial bold, 14.

- The grid displaying and adjustment is deactivated but its value is 8.
- The page is named "New page" and its type is standard, being this the only page type. Page resolution is 640x480 pixels.

Figure 3.1 depicts its graphical aspect.

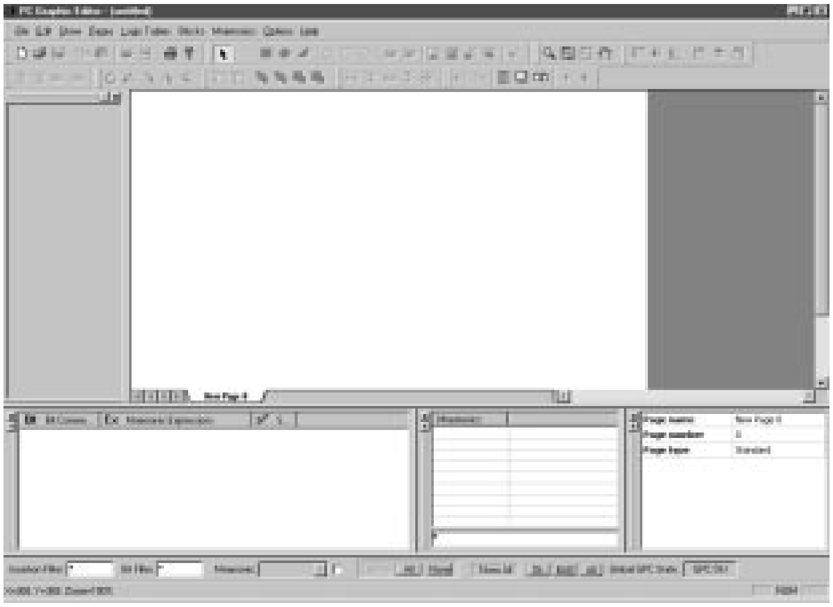
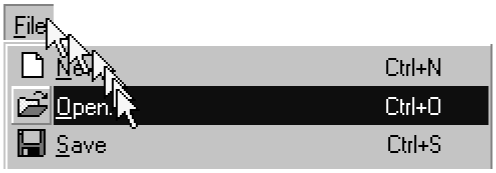


**Figure 3. 1 — New design**

### 3.1.2 Open a design

Open a GPC file version format 2.0. This operation can be done in two different ways:

1. Using menu option *File/Open*.

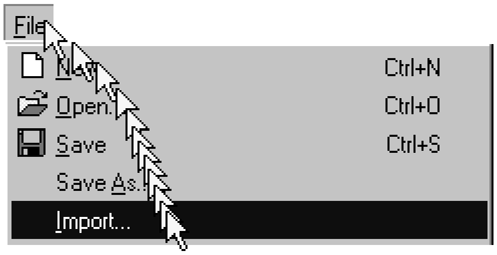


The picture shows that a shortcut Ctrl+O is also associated to this action.

2. By means of toolbar button *Open* in *default toolbar*

### 3.1.3 Import a design

The PCGE Tool imports GPC files 1.07 and 1.08. This operation can be done using menu option *File/Import…*



This operation open a gpc file version format 1.07 or 1.08 and convert it, in memory, in format 2.0. Only version format 2.0 gpc files will be saved.

## 3.2 Creating and editing blocks

### 3.2.1 Creating a Block

Blocks are essential elements into a graphic design and their correct construction from the beginning may simplify the whole design.

Blocks are composed by entities. Entities are single graphical elements (rectangles, circles…). Entities are the smallest piece of information in PCGE, they owe a set of properties and characteristics.

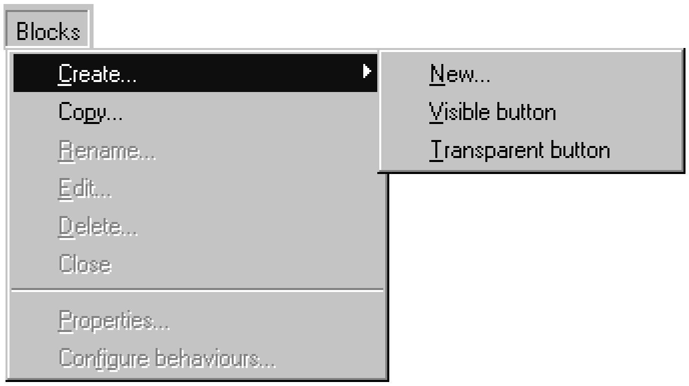
There are two kinds of blocks: button and no button. The main difference between these two types is that button blocks can have associated a set of commands when they are placed as insertions. Those sets of commands consist on a collection of integer numbers called DGRA Options in GPC format.

Button blocks can also be Transparent and Visible. Buttons aim is to have a way to program commands sending when using PCGE, this is necessary in graphics for some systems which send commands such as SCC for example. To meet this requirement, visible buttons are elements intended to be seen by the user in the target system. Typical visible buttons may be OK, CANCEL, EMERGENCY OVERRAY, etc. Transparent buttons are intended not to be seen in target systems. Typical transparent buttons could be those placed inside signal elements to provide “active screen” systems.

So far, buttons are only used in DIMETRONIC graphics because DGRA Options match with NOI options generated by DIMETRONIC Command Tool HORD.

Visible button based insertions need to determine whether insertion is pushed or released, but transparent button based insertions do not.

In order to create a button block, select *Blocks/Create* and choose the button block type desired.



In order to simplify block edition, default zoom factor is x400%.

### New

This option creates a new block reference for edition. From this moment on, user can use all *Library Toolbar* buttons to create and edit the new block.

Example 3.2 — 1: platform

| | |
|---|---|
| Entities to use: | Filled Rectangle |
| Number of entities: | 5 (4 same) |

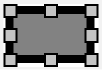
Process:

- Click on button *Filled Rect* in Block Toolbar. ▣
- Move mouse pointer to the block edition view to start drawing the rectangle.

  Note that mouse pointer has a cross aspect. +

  Click on desired point and drag the mouse to outline a rectangular shape. Release left button when rectangle size is OK.

When releasing, the entity is created.



Entity position and dimensions usually do not match with desired so parameters modification is required. For the example, the entity's characteristics are:

Entity name: AE@1 (Default) Width: 8
X Origin: 0 Height: 5
Y Origin: 0 Base width type: Fixed length

- Platform has three more entities as the previous one, so they can be copied from the original.

  (Ref. *3.5.1 Copy / Paste*)

- Once created, entities are placed in their correct position.

*Copy 1:*

Entity name: AE@2 (Default)
X Origin: 16 (Default)
Y Origin: 0

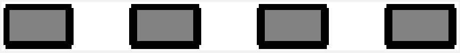
*Copy 2:*

Entity name: AE@3 (Default)
X Origin: 32 (Default)
Y Origin: 0

*Copy 3:*

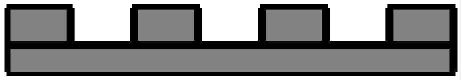
Entity name: AE@4 (Default)
X Origin: 48 (Default)
Y Origin: 0

After that, block appearance resembles to the figure:



- Finally, it is necessary to create a last entity below the previous. Parameters of the last entity could be:

Entity name: AE@5 (Default) Width: 56
X Origin: 0 Height: 4
Y Origin: 5 Base width type: Fixed length

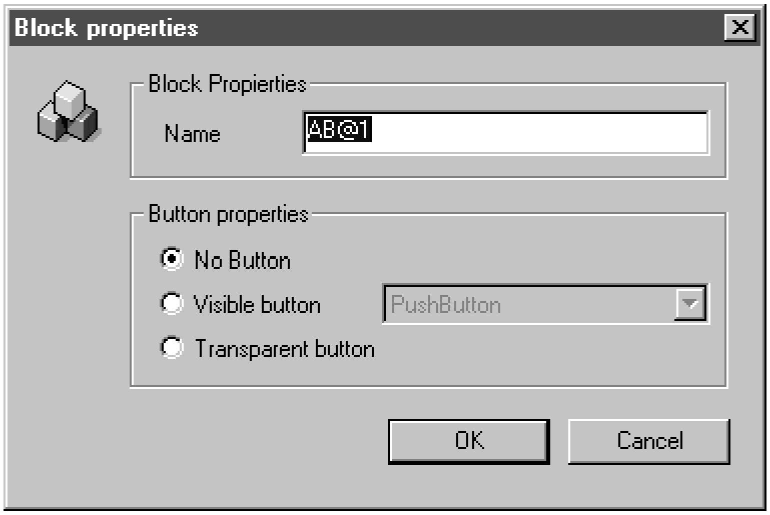


Once realised all modifications to the block entities, user must validate changes to get them updated in PCGE.

Click on Save Changes ✓ button to validate changes. This button is in *Library Toolbar*. When a block is first validated after creation, a dialog prompting for a block's name is launched.

This dialog it is divided into two parts: Block Properties and Button properties. In the first area it is shown the default block's name assigned by PCGE. This default name consists of a literal part fixed as AB@ and a variable part consisting of a number. I.e. AB@1.

Second area controls block's type, button or no button type. By default block is no button. Next picture depicts block's properties dialog.



Block properties

Block Propierties

Name: AB@1

Button properties

● No Button

○ Visible button

○ Transparent button

**Figure 3. 2 — Block Properties Dialog**

Once validated, block is inserted in the library window and consequently is ready to use.

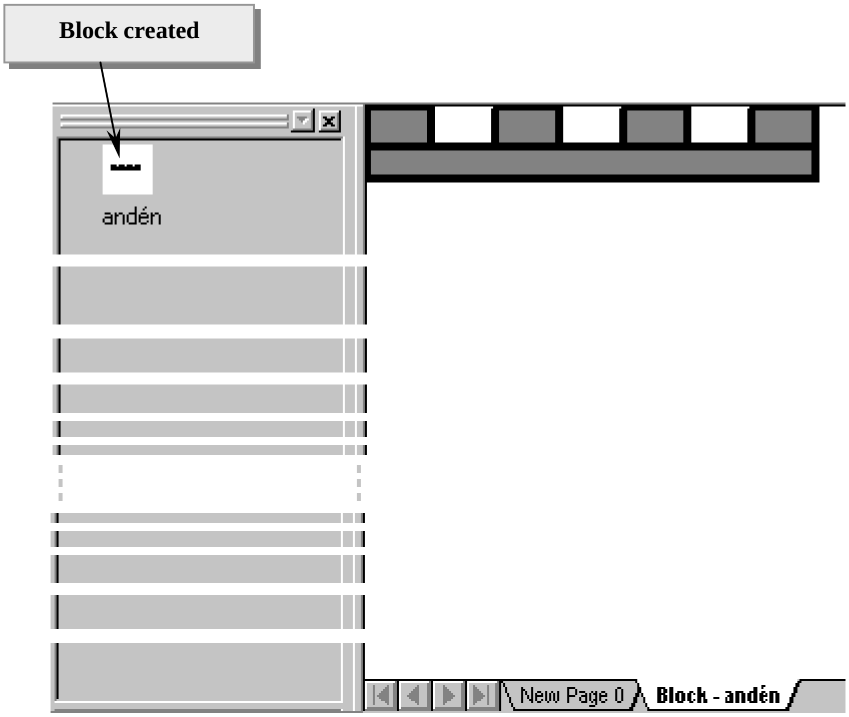


**Figure 3. 3 — Block Validated**

When generating a non-button block, PCGE assigns a *Default Behaviour* automatically for it. The default behaviour has a set of default logic tables consisting in constant LT’s with the default colour.

(See Ref. *3.3 Creating and Editing Logic Tables*).

### Visible Button

Another feature in PCGE is creating visible buttons. This is a special type of block. A visible button consists of a FillRect, an EmptyRect and a Text.

Its graphical aspect is shown in figure 3.4.

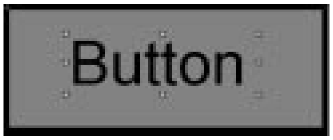


**Figure 3. 4 — Visible Button**

PCGE automatically generates two default behaviours for a visible block, called *Default Behaviour Up* and *Default Behaviour Down*, assigning a set of default LT's to each one. User can modify the LT's associated to the behaviours, but he will not change or rename the behaviours.

(See Ref. *3 Creating and Editing Logic Tables*).

User also can add new entities to a visible block button, or edit those default entities.

### Transparent Button

It is required that lies on the need of executing command when clicking inside an insertion in active screen systems, as SCC for example.

When a transparent button is created, an EmptyRect entity is automatically created. User can add new entities or modifying existing.

PCGE automatically assigns a default behaviour for transparent buttons *called Default Behaviour Up-Down*. This behaviour cannot be changed or deleted, although its set of LT's can be modified.

Next picture shows a transparent button as it is created by PCGE.

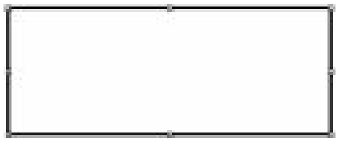


**Figure 3. 5 — Transparent Button**

Button blocks are presented with the special character '>' when placed in Library Window.

### 3.2.1.1 Entities

Entities are the simplest graphical elements in PCGE. A block is composed by a set of entities.

Next type of entities are available:

**Entity** | **Associated button**

- *Line*: Creates a line. *Shift* key pressed when creating a line allow **ORTO** mode. ⟋

  Properties of lie are:
  - · Entity name must be unique inside its block.
  - · X Origin, Y Origin
  - · X End, Y End
  - · Width.

  In order to draw a line just click on origin point and drag until end point.

- *Filled Rect*: Generates a filled rectangle. If *Shift* key is pressed, then the rectangle becomes a square. ■

  Properties:
  - · Entity name must be unique inside its block.
  - · X Origin, Y Origin; TopLeft point of the rectangle.
  - · Width.
  - · Height.
  - · Base width type; defines the type of resizing of the entity when inserted in the design area: fixed, right or left.
  - · Display Fill; specify how show the fill rectangle. If *Yes* option is selected, the entity will be shown like as fill rectangle, while if *No* options is selected it will be shown like as empty rectangle. This option can be useful when drawing elements for marking insertions according to systems based in [1].

  In order to draw a Filled rectangle click on origin point and drag until end point.

- *Empty Rect*: Generates a hollow rectangle. If *shift* key is pressed then a hollow square is generated. □

  Properties:
  - · Entity name must be unique inside its block.
  - · X Origin, Y Origin; TopLeft point of the rectangle.
  - · Width.
  - · Height.
  - · Line width.

  In order to draw a Filled rectangle click on origin point and drag until end point.

- *Filled Circle*: Generates filled ellipses. If *shift* key is pressed then a circle is generated. ●

  Properties:
  - · Entity name must be unique inside its block.
  - · X Origin, Y Origin; TopLeft point of the bounding rectangle.

· Width of the bounding rectangle.
· Height of the bounding rectangle.

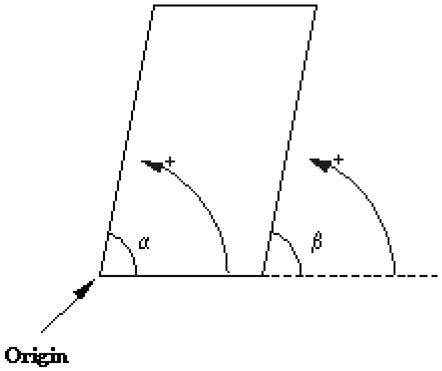
- *Empty Circle*: Generates hollow ellipses. If *shift* key is pressed then a hollow circle is generated.

  Properties:
  · Entity name must be unique inside its block.
  · X Origin, Y Origin; TopLeft point of the bounding rectangle.
  · Width of the bounding rectangle.
  · Height of the bounding rectangle.
  · Line Width.

- *Parallel*: Creates parallelograms.

  Properties:
  · Entity name must be unique inside its block.
  · X Origin, Y Origin; bottom-left point of the parallelogram.
  · Base Width; base of the parallelogram.
  · Height of the parallelogram.
  · α, β. Angles defined by the edges of the parallelogram. Always positive (although PCGE can read negative values from older GPC files for compatibility).
  · Height width Type. Type of resizing of the entity when inserted in design view. It can be fixed, top or bottom.

  Next figure depicts a parallelogram with α and β positive.



- *Polygon:* Generate a hollow closed polygon with straight edges.

  Properties:
  · Entity name must be unique inside its block.
  · X Origin, Y Origin; TopLeft point of the bounding rectangle.
  · Width of the bounding rectangle.
  · Height of the bounding rectangle.
  · Line Width.

In order to draw a polygon, just click on different points it is composed. A vortex is added when you click over a point. Double-click on the last point to finish drawing.

- *Text*: Text string determined at entity level. It is a constant text unless it is edited.

  Properties:

  · Entity name must be unique inside its block.

  · X Origin, Y Origin bottom-left point of the bounding text rectangle..

  · Font Name. Name of the font used for the text. By default is used Default Font (Ref. *3.1 Creating a New Block*).

  · Text String.

- *Attribute Text*. Text fixed to a entity level but editable to a insertion level.

  Properties:

  · Entity name must be unique inside its block.

  · X Origin, Y Origin bottom-left point of the bounding text rectangle.

  · Font Name. Name of the font used for the text. By default is used Default Font (Ref. *3.1 Creating a New Block*).

- *Bitmap — Edit Bitmap:* Creates a monochrome bitmap in which user can create its own images. First button is used to create an entity bitmap as if a rectangle. Once done, second button enters edition mode for that bitmap. An auxiliary grid is shown to let the pixel-by-pixel drawing. Left mouse button sets a pixel, right mouse button unselect a pixel.

  These entities cannot be rotated and so do not insertions containing these entities.

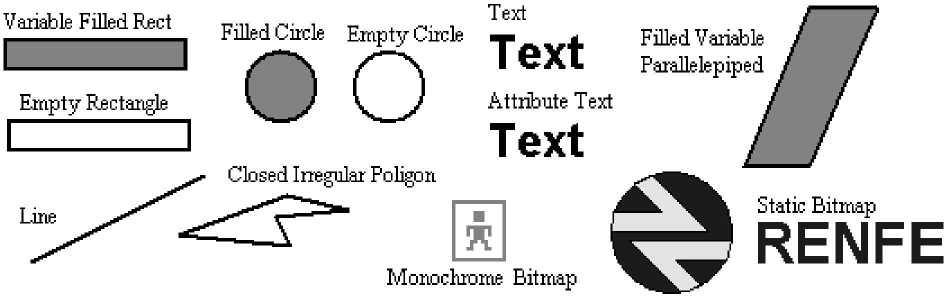
- *Static Bitmap — Import Bitmap*: Creates a bitmap of 256 colours. It can be created using two different ways:

  ▪ By means of second button, that becomes active when exists a static bitmap entity inside block. User can select a 256-colors file for importing.

  ▪ Through clipboard if a 256-colors bitmap is inside it. Then, paste command places the bitmap in PCGE.

  These entities cannot be rotated and so do not insertions containing these entities.

The next picture shows an example of each kind of entity.

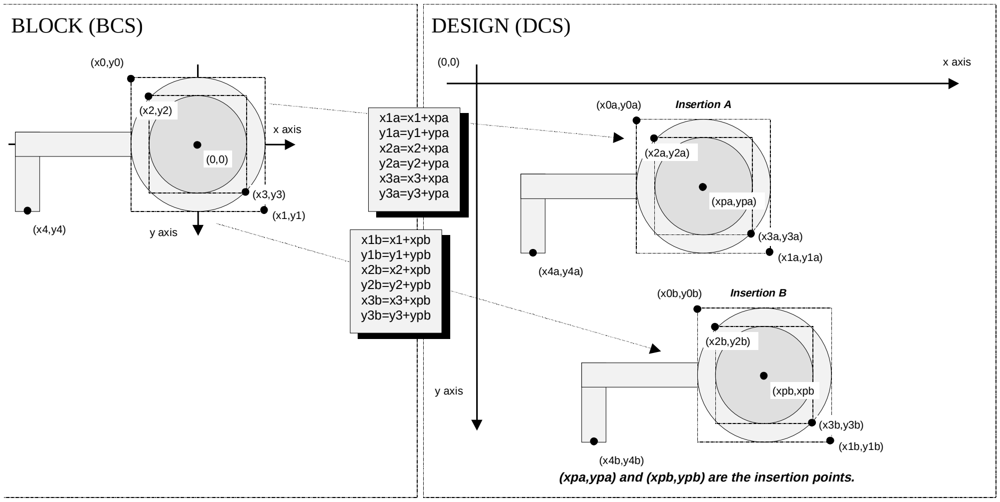


As explained before, entities are grouped in blocks to be represented on the design. At this point it is important to define two different co-ordinate systems:

Block Co-ordinate System (BCS). This is the co-ordinate system reference to all entities in a block. The co-ordinate origin is the so-called insertion point in the block, used by the PCGE tool to reference the insertion location in the design.

Design Co-ordinate System (DCS). This is the co-ordinate system used to reference insertions and labels in the design.

Because entities inside a block are defined in BCS, and the insertion using it are defined in DCS, a co-ordinate conversion is needed to translate BCS points into DCS:



Then a design co-ordinate (xp, yp) can be defined:

$$xp = xe + xpi$$

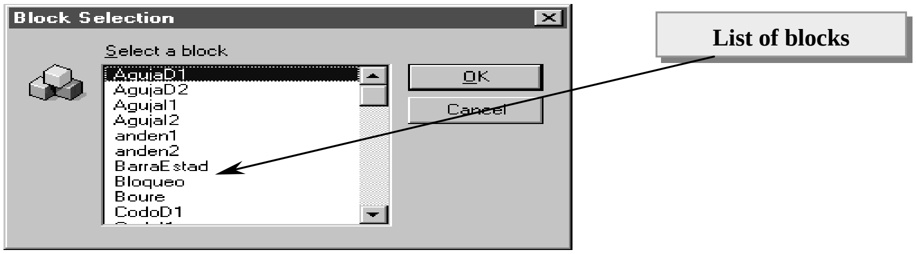
$$yp = ye + ypi$$

where (xe, ye) is the entity co-ordinate in BCS and (xpi, ypi) is the insertion point in DCS.

### 3.2.2 Editing a Block

Editing a block is as important as creating it. PCGE allows user to modify a block at any time regardless of the block is being associated by insertions or not.

- One way to edit a block is through by means of the menu option *Block/Edit…*, which opens a dialog prompting for a block to edit from a list of blocks available. This list is alphabetically shorted showing first non-button blocks and button blocks at the end of the list.



AgujaD1
AgujaD2
Agujal1
Agujal2
anden1
anden2
BarraEstad
Bloqueo
Boure
CodoD1

List of blocks

**Figure 3. 6 — Block Selection (Edit a Block)**

- There is a reference by block inside Library View. A block can be edited from this window by double clicking on the picture or with contextual menu Edit (right mouse button).

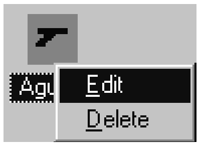
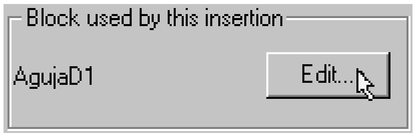


**Figure 3. 7 — Contextual Menu (Edit a Block)**

- When opening the insertion's properties dialog, a button for associated block edition let open the insertion's associated block for edition through *Edit* button.



AgujaD1

**Figure 3. 8 — Insertion Properties (Edit a Block)**

Once opened a block for edition using whichever of the aforementioned ways, user can modify the block adding entities, changing the set of LT's for a behaviour, etc.

User must validate changes done in a block so that they are registered. For that purpose he must press validation button ✓ in Library Toolbar. Validation button causes a dialog to introduce block name when block is just created. Once created block, validation button only validate changes.

When a block is validated, all associated insertions in design view are updated.

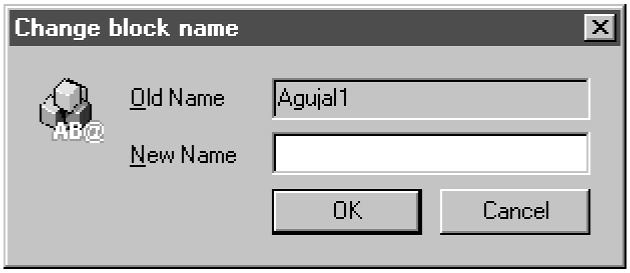
### 3.2.3 Other Operations with blocks

There are other operations available on blocks.

**Block/Copy…**

This action copy the block selected from the list (see fig 3.6) with other name.

Figure 3.9 shows the dialog displayed when copy option is chosen.



Old Name: Agujal1

**Figure 3. 9 — Copy a block**

**Block/Rename…**

Allow renaming an existing block.

**Block/Delete…**

Only blocks without associated insertions can be deleted from a design. When selecting this option a list of blocks is displayed (see figure 3.6), from which they can be selected for deletion.

Contextual menu **Delete** from *Library View* can also be used for deleting a block.

***Note***

Operations that produce an element deletion will require user confirmation.

**Block/Properties…**

This option allow change a block's properties after selecting a block from the list of available blocks (see fig 3.6). A Block property dialog is shown in fig 3.2, inside it is possible to modify block type, converting a block from button to non-button and vice versa.

**Blocks/Configure behaviours…**

This menu option is active when an entity is selected. It also can be accessed by double-click on a particular entity.

By this option, user can assign a Logic Table to one particular behaviour for each entity.

Using the *Configure Dialog* dialog, user can create, delete and rename behaviours inside the selected block. Default behaviours are automatically created by PCGE when creating a block.

Entities belonging to a no-button block own a *Default Behaviour* when created.

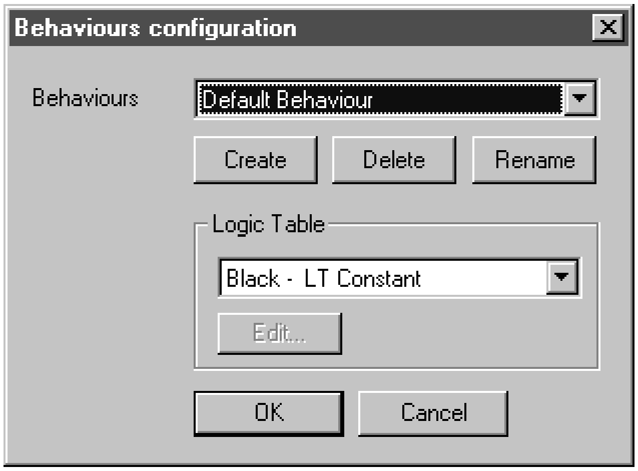
Entities belonging to a visible-button block owe a *Default Behaviour UP* and *Default Behaviour Down* when created.

Entities belonging to a transparent-button block own a *Default Behaviour UP-DOWN* when created.

Entities in button-blocks (either visible or transparent) only can have the aforementioned behaviours, i.e. they cannot be deleted or renamed. No other behaviours can be created either.

All entities have assigned a default LT for its default behaviour when first created. This LT is a constant LT of the colour chosen as default colour. Inside block must be at least one behaviour and each entity must have a LT for each LT, except Static Bitmap entities, that must not have associated LT.

Figure 3.10 shows the dialog of behaviour configuration.



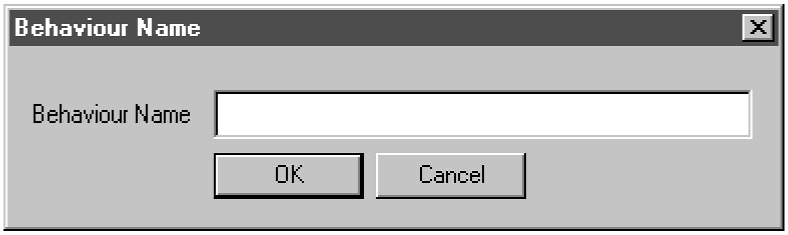
Behaviours configuration

Behaviours: Default Behaviour

Logic Table: Black - LT Constant

**Figure 3. 10 — Behaviours Configuration (no bottom block)**

Click on **Create** button to create a new button for that block. This action will display a dialog to introduce the new behaviour’s name.



Behaviour Name

Behaviour Name

**Figure 3. 11 — Create a behaviour**

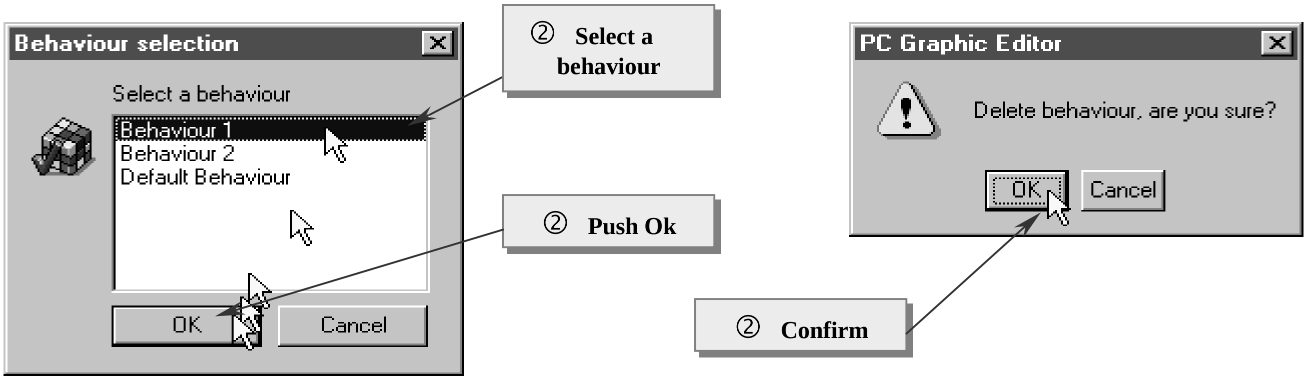
Click on **Delete** button if you want to delete a behaviour. This action will display a list of behaviours for that block. Selection is single. The behaviour will be deleted only if it is not being used by any insertion. Figure 3.12 summarise those steps.

**Figure 3. 12 — Delete an unused behaviour**

## 3.3 Creating and editing Logic Tables

Logic tables define the state of an entity at real time in the client system.

There are two types of Logic Tables: classic and extended. Both are created and edited in PCGE using menu option *Logic Tables/Edit…* or from the Behaviours Configuration dialog explained in the last section (see fig 3.10).



② Select a behaviour

② Push Ok

② Confirm

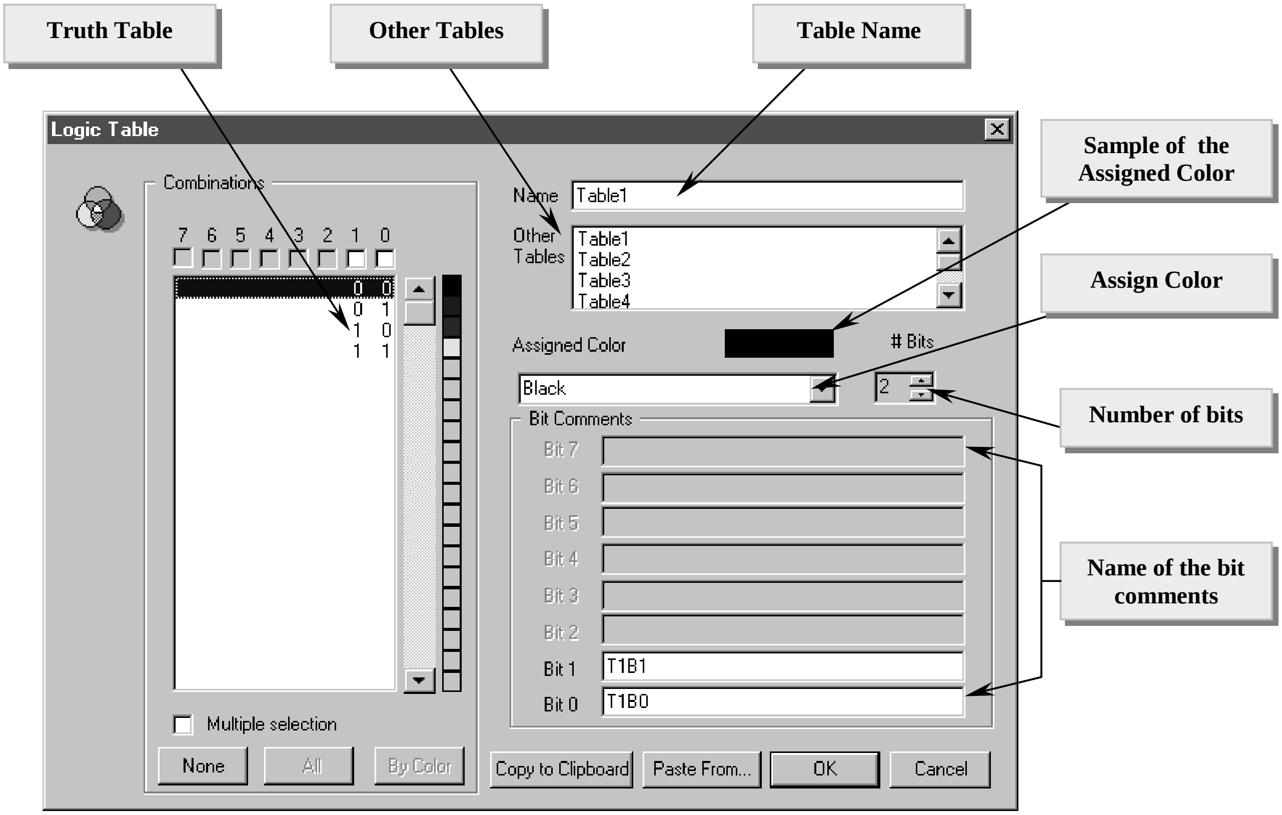
### 3.3.1 Classic Logic Tables

A Classic Logic Table is similar to a truth table in digital design. Maximum number of bits that PCGE manages in CLT's is 8, therefore the maximum number of possible combinations in CLT's is $2^8$ = 256. This restriction is for compatibility reasons with a former Graphic Tool used in DX (HPAN-HGRA).

One LT defines a colour by each possible combination of n bit comments defined. In total $2^n$.

Select menu option *Logic Tables/Create/Classic…* in order to create a Classic LT. See figure 3.13 to see the aspect of a CLT.

**Figure 3. 13 — Classic Table Logic**



Name: Table1

Other Tables: Table1, Table2, Table3, Table4

| | |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

Assigned Color: Black

# Bits: 2

Bit Comments

Bit 7:
Bit 6:
Bit 5:
Bit 4:
Bit 3:
Bit 2:
Bit 1: T1B1
Bit 0: T1B0

Dialog is divided in two areas. Left area shows the truth table generated according to the number of bit comments. Column at the right shows the colour assigned to each combination.

Multiple selection check box allow multiple selection of combinations to simplifying colour assignation when several combinations have the same colour assigned. When this mode is active, clicking one combination with mouse do not unselected other combinations already selected.

Button **None** unselected all combinations selected at that moment. Buttons **All** and **By Colour** are active in multiple selection mode. The first one select all combinations in the table, the second selects all combinations that share the same colour than the active combination selected.

Right area of this dialog manages name of bit comments and colours. In the top area is stored the name of the LT. Just below a list of all classic LT's available in the design. This last control is only informative.

Below the list control are the controls related with the LT bit comments and the colour selector.

Finally, some command buttons. **Copy to clipboard** copies the LT content to the clipboard so that it can be easily printed or copied to a other document. **Paste from…** button let copy the contents of other LT inside the current LT (bit comments and colours). The other buttons let validate and cancel. The example 3.3 — 1 shows the aspect of a text file whose contents have been pasted from the clipboard.

Example 3.3 — 1: Information of a Logic Table in to Clipboard

```
Logic Table Name: Table1

Combinations:

7 6 5 4 3 2 1 0

---------------
            0 0 Black
            0 1 Blue
            1 0 Brown
            1 1 Yellow


Bit Comments:
---------------
Bit 7:
Bit 6:
Bit 5:
Bit 4:
Bit 3:
Bit 2:
Bit 1: T1B1
Bit 0: T1B0
```

Classic Logic Tables have two disadvantages:

- 8 bits restriction.
- Difficult to create and edit when number of bit comments is higher than 5.

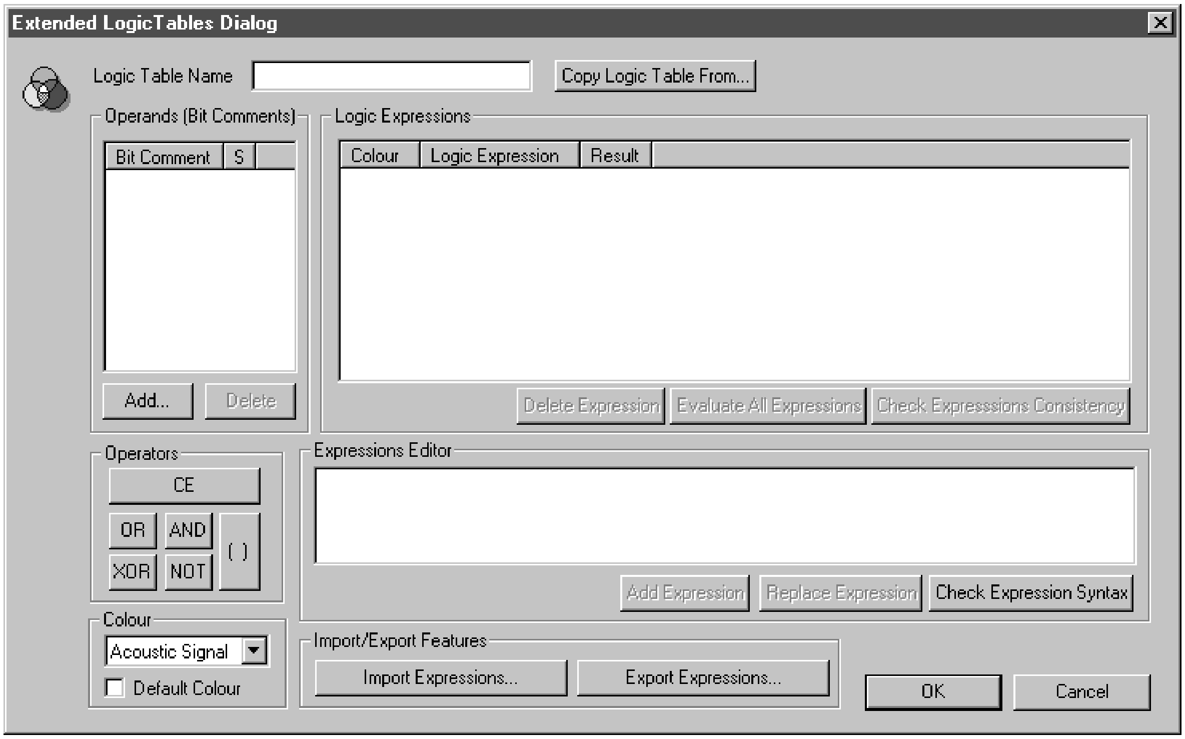
To overcame those disadvantages, Extended Logic Tables were created.

### 3.3.2 Extended Logic Tables

Use menu option *Logic Tables/Create/Extended…* in order to create an ELT.

This type of table use booloean expressions, which permits to represent every possible logic combination. This way, disadvantages in CLT's are not in ELT's.

ELT's managing is simple. Figure 3.14 shows the aspect of one of this tables.

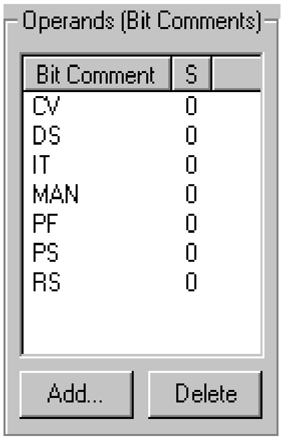


Colour: Acoustic Signal

□ Default Colour

**Figure 3. 14 — Extended Table Logic dialog**

As in CLT's, in the upper area of the dialog user can enter the name of the ELT. Button **Copy Logic Table From…** let copying information from other ELT. If this button is pushed the logic table selection dialog will be shown with available Extended Logic Tables.



| Bit Comment | S |
|---|---|
| CV | 0 |
| DS | 0 |
| IT | 0 |
| MAN | 0 |
| PF | 0 |
| PS | 0 |
| RS | 0 |

**Operand Bit Comment** is the area where the operands used to build logic expressions. These are Bit Comments in old Logic Tables PCGE application.

*Bit Comment:* operand. Click with left button on Operand name to change it name (the same as in Windows explorer right panel). Press 'Supr' button in keyboard after selecting an operand to delete it. Double click with left button on Operand name to add it to current expression being edited in expression Edition Edit Box. Click with right button on Operand name to change Operand State.

*S*: operand current state. This can be 0 or 1.

*Delete:* delete selected Operand (Bit Comment)

*Add:* add a new Operand (Bit Comment) to ELT. Click to open the next dialog where the name and the operand desired state can be defined. After click Ok the information is placed in operands frame alphabetically shorted.

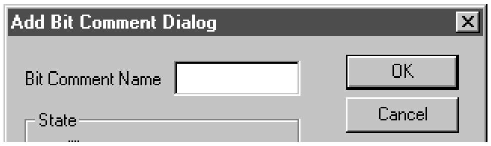
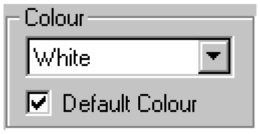


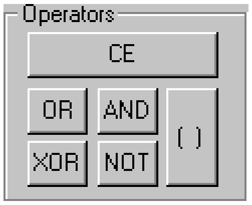
**Figure 3. 15 — Add Bit Comment dialog**


Colour
White
☑ Default Colour

This control assigns the equation inside the expressions editor to selected colour. There must allways be a default colour for the ELT. The default colour of the ELT has to be unique. As well the check box Default Colour is the colour used to represents elements in client system, when the evaluation for each logic expression is 0.

**Operators** .

Click with left button to add Operator to current expression being edited in Expression Edition Edit Box.



*OR:* OR Gate. Use: [Operand1] **OR** [Operand2]

*AND:* AND Gate. Use: [Operand1] **AND** [Operand2]

*XOR:* XOR Exclusive OR Gate. Use: [Operand1] **XOR** [Operand2]

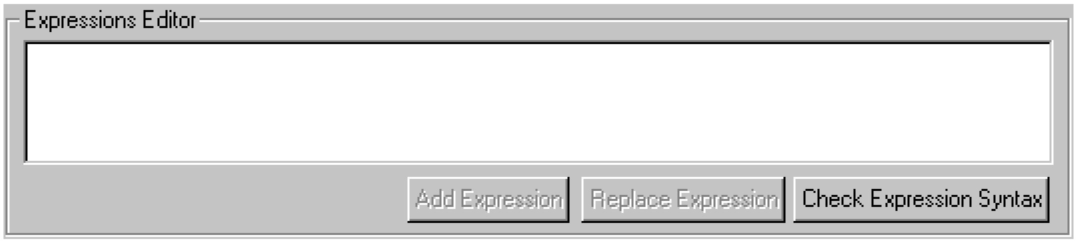
*NOT:* NOT Gate. Use: **NOT** [Operand1]

*( ):* Brackets. When clicking on the button both brackets (open and close) are added to the expression being edited. All the Operands and Operators added after will be inserted inside brackets.

*CE:* Clear Expression Button. Delete Expression being edited.

**Expression Editor**

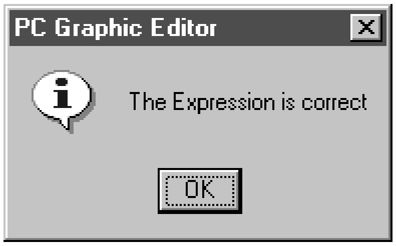
Expression editor allows user to construct a boolean expression with all necessary operands and operators. In order to introduce an expression double clicking on the desired bit comment (bit comment = operand) and single click on the logical operator (OR, AND, etc). Repeating these actions the boolean expression will be composed.



*Add Expression:* add the expression, which is being edited to the Expression List. Click the button to add the expression to the list. The associated colour to the expression will be equal to the one currently selected in the Colours Combo Box. If the expression syntaxes are not correct an error message will be shown.

*Replace Expression:* Replace the expression selected in the Expressions List with the one being edited.

*Check Expression Syntax:* Button used to check the syntaxes of the expression being edited. Click the button to check the syntaxes. A message Box will be shown with the checking result. If expression is correct a dialog is shown.



The Expression is correct

**Logic Expression**

Full list of Logic Table Expressions. There are three columns:

*Colour:* Result Colours for each Expression. This is the colour that will be shown in client systems when the associated expression evaluation result is 1. To change the colour for a specific expression, select it from the list and then use the Colours Combo Box to select the new desired colour.

*Logic Expression*: Expression used to define the logic associated to a specific colour. To perform an operation on an expression, first select it from the list and then use Expressions Management buttons to replace or delete it. In addition, an expression can be deleted selecting it and pressing 'Supr' button on the keyboard. In order to edit an expression, double click on it. It will be directed to the expression edit box.

*Result*: Result of the evaluation of each expression using current states defined for each operand.

There are three control buttons:

*Delete Expression:* Delete the expression selected in the Expressions List.

*Evaluate All Expressions:* Evaluate all the expressions stored in the Expressions List, by using the current state defined for each Operand from the Operands List. Click the button to evaluate all the expressions, The result will be shown in Expressions Result List. Note that only expression can be evaluated as 1 for a specific Operands combination.

*Check Expression Consistency:* Evaluate all expressions for all Operands combinations checking that only one of them is evaluated as TRUE for each combination. When the button is clicked a progress dialog with a cancel button is shown. The process can be stopped at any time clicking the cancel button. An estimated process time is also shown as well as the number of evaluation which is being carried out. Note that the estimated time is only an indication and it depends on the machine and the system load.

**Import/Export Features**

Expressions Import/Export operations which can be carried out.

*Import Expressions…:* Import a List of Expressions from a text file.

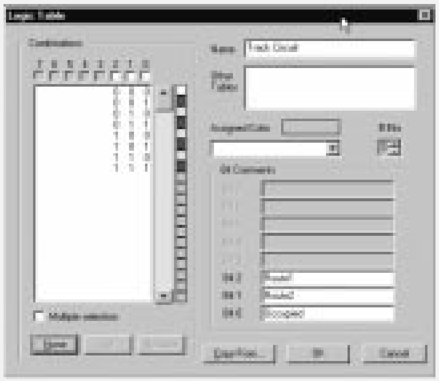
*Export Expressions…:* Export all the Expressions stored in Expressions List to a text file.

The format of the Expressions file must be:

- Each Expression must be defined in a row.
- There can’t be blank lines.
- Each Expression will have the format: ‘Colour’ = ‘Expression’

Example 3.3 — 2: Extended Logic Table from Classic Logic Table

Suppose a classic Track Circuit Logic Table:



Name: Track Circuit
Bit 2: Route1
Bit 1: Route2
Bit 0: Occupied

To create the equivalent Extended logic table using Boolean expressions follow the next steps:

*Open Extended Logic Table Dialog*

*Click ‘Logic Tables/Create/Extended…’ menu option*

*Create Operands (Bit Comments) Occupied, Route 2 and Route* ***1****.*

1. Click on ‘Add Bit Comment…’ button. The Add Bit Comment dialog will be shown.
2. Input ‘Occupied’ in Bit Comment  Name field.
3. Press OK
4. The operand will be added to Operands List
5. Repeat the process for the rest of Operands

*Create Logic Expressions for White, Red and Yellow Colours*

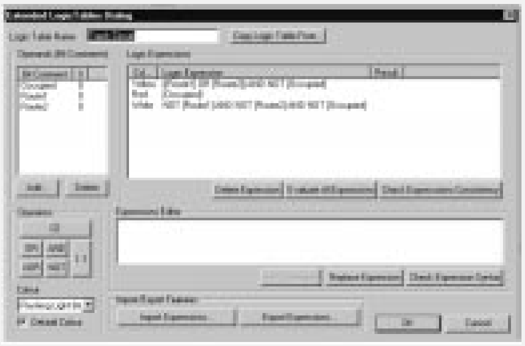
1. For Yellow Colour the expression is: (Route1 OR Route2) AND NOT Occupied.

2. Click ( ) button.
3. Double Click on Route1 name in Operands List.
4. Click on OR button.
5. Double Click on Route2 name in Operands List.
6. Click on Expression Edition Edit Box after close bracket.
7. Click on AND button.
8. Click on NOT button.
9. Double Click on Route2 name in Operands List.
10. Select Yellow Colour in the Colours Combo Box.
11. Click on 'Add Expression' Button.
12. The Expression with it is represented in *Logic Expression* List
13. Repeat the process for the following expressions:
    NOT [Route1] AND NOT [Route2] AND NOT [Occupied] *-> White colour*
    [Occupied] *-> Red Colour*

*Evaluate All Expressions and check its consistency*

1. Select a combination for Operand States. For example Occupied=0, Route1=0, Route 2=0. To do that, right click on Route 1 name in Operands List to change State between and 1. Try the same for Route 2.
2. Click on Evaluate All Expressions Button.
3. The result must be White=0, Red=0 and Yellow=1 (as it is shown in the following figure).
4. Try any other combination and check results.

Click Check Expressions consistency button. A dialog will be shown telling that the expressions are consistent. This means that for each operand combinations only one expression is evaluated to TRUE.



## 3.4 Behaviours

Generally speaking, many blocks with the same graphic appearance only differ, once inserted, in the logic tables used for its entities[1]. To avoid repeating multiple blocks, and because a block is the most adequate object to group entities, a link with a set of logic tables is defined, instead of using particular logic tables. This set is called *behaviour*.

[1] This grouping level is not aviable in the DGRA file format. This new quality save a lot of memory, avoiding repeating the same logic tables in the design.
Another advantage of this method is the library simplicity, because the user do not need to repeat blocks with only differences in its associated logic table.

Using behaviours, the desired one is selected at insertion level, and the block restrict the set of possible behaviours to select.

A sample of this matter is shown on the next page: A block with two different behaviours assigned. Both of these behaviours uses the *Logic Table 1* for *Entity 1*, but *Behaviour 1* assigns *Logic Table 2* for *Entity 2*, and *Behaviour 2* assigns *Logic Table 3* for *Entity 2.*

Note that this logic tables can be Normal Logic Tables or Extended Logic Tables.

If the insertion is a button insertion the client must select two behaviours:

- DOWN behaviour. It is used when button insertion is pressed.
- UP behaviour. It is used when button insertion is not pressed.

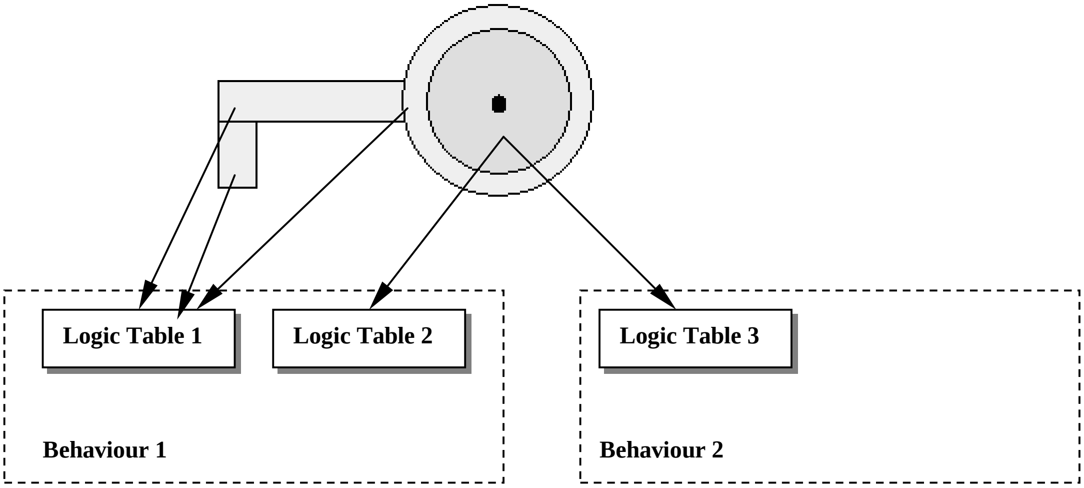


**Figure 3. 16 — One block with several behaviours associated**

## 3.5 Creating and editing insertions

Insertions are design elements. They are basically block instances whose main objective is to render its associated block in a design.

Insertions do not own any information about entities that they are composed, instead, they only own information about their insertion point and a reference to their associated block.

An insertion not only stores graphical data, but they own the necessary information to configure its behaviour according to the associated LT's to their block's entities.

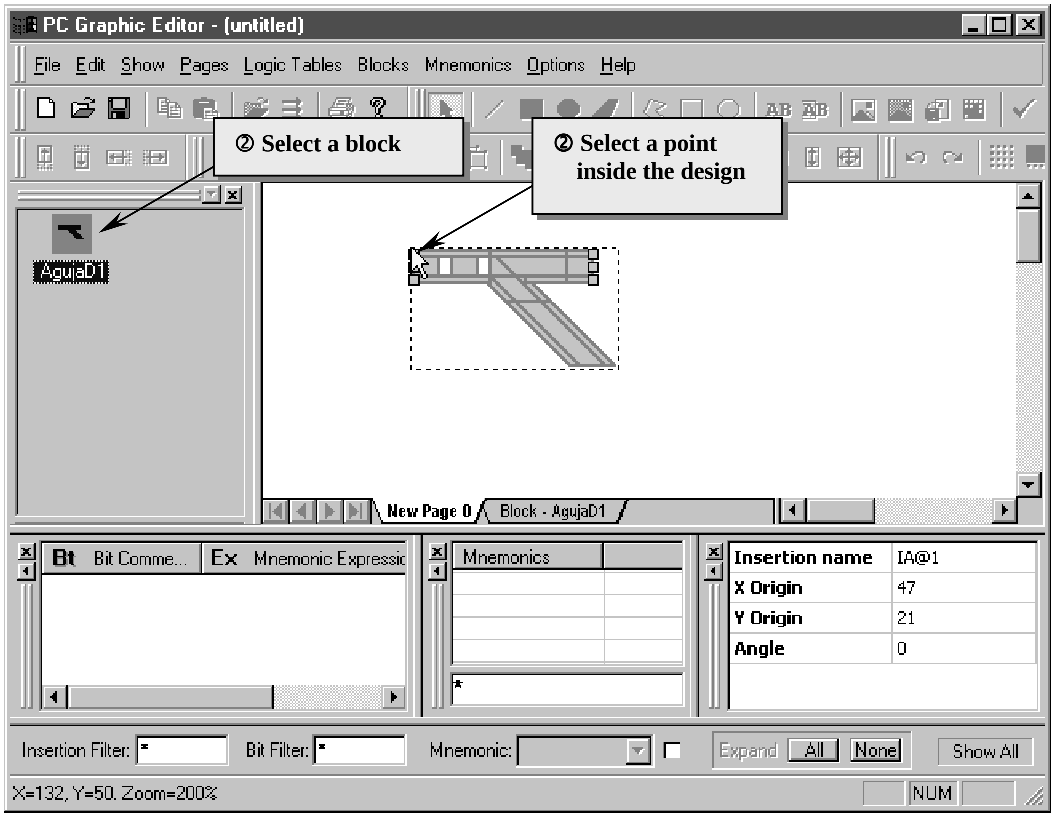
There are a kind of insertions that are able to execute commands, those are which are button instances. Client system can execute commands when those insertions are pressed.

In order to create an insertion proceed in the following way:

1. Select block from which insertion will be instantiated from Block Toolbar.

2. Click on a point inside design view area.

3. A new insertion is now added to the design.



② Select a block

② Select a point inside the design

| Insertion name | IA@1 |
|---|---|
| X Origin | 47 |
| Y Origin | 21 |
| Angle | 0 |

X=132, Y=50. Zoom=200%

**Figure 3. 17 — How to create an insertion**

In the figure 3.17, it can be seen how a new insertion is generated and its properties are shown in the properties view.

Each new insertion is auto-named by PCGE with a default name. It begins with the prefix IA@ and followed by a number. Insertion's name along with the rest of insertion's properties can be modified at any moment.

Insertion's properties edition can be easily made using properties window.

Prior to continue speaking about insertions, an additional explanation on different bit comments types is required here.

All non-constant entities have associated bits to define their behaviour in the installation. Constant entities have associated zero bits, so only one entity status is possible at any time.

From the user point of view, a bit is defined by a comment. From the system, a bit is defined by its location, this is, a method used to address it. The GPC file format allows three different kinds of bit location:

1. *Direct Bits*: From one to five integer numbers to define its location in the field. Usually, those numbers are associated with System, Line, Housing, Card, Bit, but can be customised to any other configuration. SSI uses this type of bits.

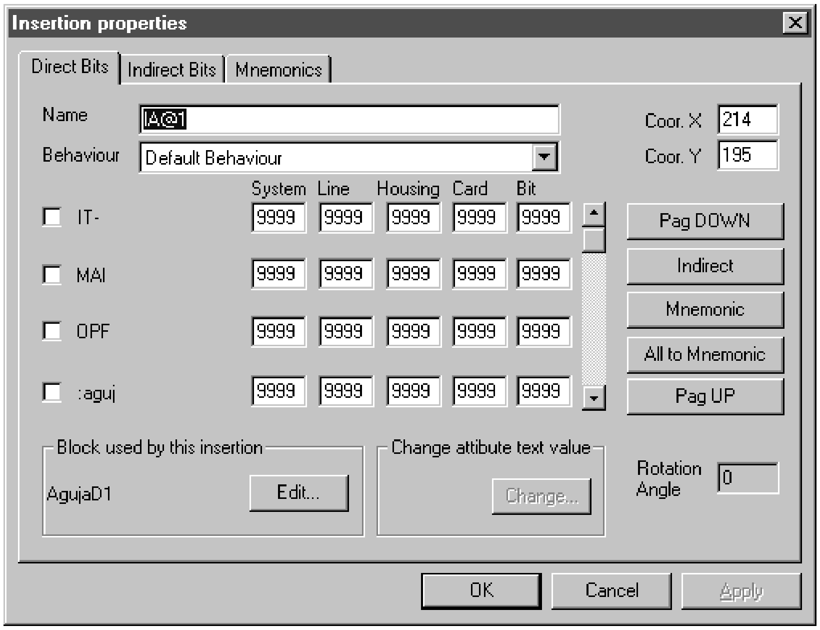
2. *Indirect Bits:* These are integer values received directly from the system, representing different status not meaningful to the field.

3. *Mnemonics:* Coded words, directly being processed by the client system, and associated with direct or indirect bits.

These three kinds of bit location can be used in the GPC file format at the same time.

Now, it is possible to go on with insertions.

Double-click over one insertion displays next properties dialog:



Insertion properties

Name: IA@1
Behaviour: Default Behaviour
Coor. X: 214
Coor. Y: 195

|  |  | System | Line | Housing | Card | Bit |
|---|---|---|---|---|---|---|
| ☐ | IT- | 9999 | 9999 | 9999 | 9999 | 9999 |
| ☐ | MAI | 9999 | 9999 | 9999 | 9999 | 9999 |
| ☐ | OPF | 9999 | 9999 | 9999 | 9999 | 9999 |
| ☐ | :aguj | 9999 | 9999 | 9999 | 9999 | 9999 |

Block used by this insertion: AgujaD1
Rotation Angle: 0

**Figure 3. 18 — Insertion Properties dialog**

This dialog is divided in three tabs: *Direct Bits*, *Indirect Bits* and *Mnemonics*. Each tab serves to configure bits according to their type. Other fields are common in all the tabs, and correspond to those properties independent from the bit type.

### Name

Insertion name.

### Coordinates

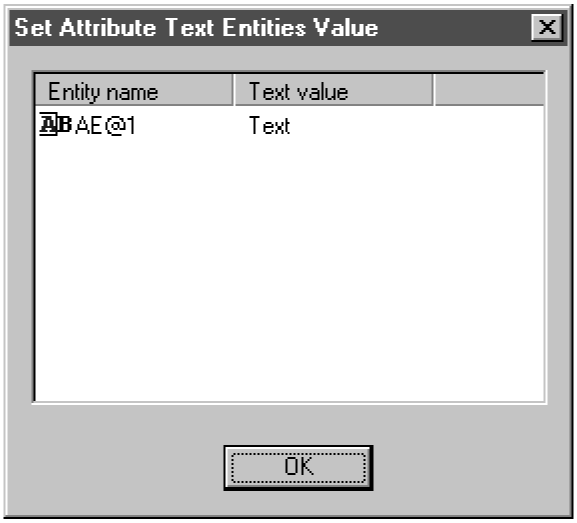
X-Y point of the TopLeft corner where insertion is placed in the design.

### Block used by this insertion

Located in the lower left side of the dialog. Using button **Edit…** block view is opened for this block edition.

### Change Attribute Text value

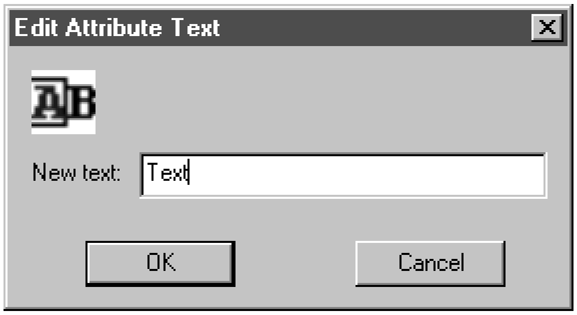
This control becomes active when insertion has AttributeText entities. **Change…** button let edit the text string.



| Entity name | Text value |
|---|---|
| AE@1 | Text |

**Figure 3. 19 — Attribute Text Entities Value dialog**

All the Attribute Text entities belonging to the insertion are listed in the dialog above. In order to modify the value of one of them, just double-click over entity name or over its current value and a new dialog to introduce its name will be displayed.



New text: Text

**Figure 3. 20 — Edit Attribute Text dialog**

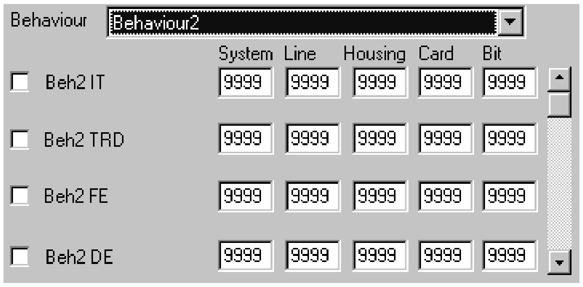
### Rotation Angle

Non editable field. Informs about he value of rotation angle.

### Behaviour

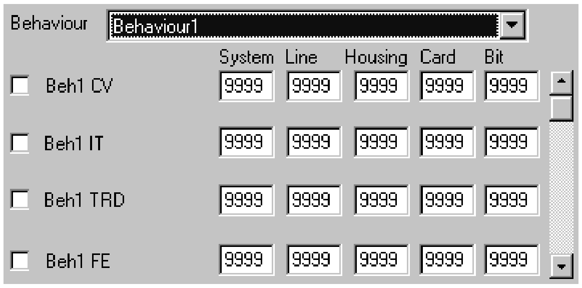
This combo contains all the behaviours associated to the block. Only one behaviour must be active per insertion. The insertion will behave depending upon the selected behaviour, i.e., depending on the set of LT's associated to the behaviour.

Next figure shows the difference between bit comments set of the same insertion with two different behaviours selected. (Direct bits).



Behaviour: Behaviour2

| | System | Line | Housing | Card | Bit |
|---|---|---|---|---|---|
| □ Beh2 IT | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ Beh2 TRD | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ Beh2 FE | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ Beh2 DE | 9999 | 9999 | 9999 | 9999 | 9999 |



Behaviour: Behaviour1

| | System | Line | Housing | Card | Bit |
|---|---|---|---|---|---|
| □ Beh1 CV | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ Beh1 IT | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ Beh1 TRD | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ Beh1 FE | 9999 | 9999 | 9999 | 9999 | 9999 |

**Tab Direct Bits**

One insertion have all of their bit comments as direct bits when it is created. In this case, bit comments must be configured according to *System*, *Line*, *Housing*, *Card* and *Bit* values[2].

Should you want to change some direct bits to indirect or mnemonics, use right side buttons. Buttons goal are (in any tab):

- Direct: Change selected bits (check box) to direct.
- Indirect:. Change selected bits (check box) to indirect.
- Mnemonic: Change selected bits (check box) to mnemonic.
- All to Mnemonic: Change all bits, regardless they are selected or not, to mnemonics.

**Tab Indirect Bit**

In this tab, bit comment configuration is focused in only one value. Those bits are used for general actions over to the system.

**Tab Mnemonics**

In this tab, bit comments are configured using mnemonics and boolean equations. As boolean operators are being inserted, the dialog checks syntactically the expression and informs about.

Mnemonics are always introduced between square brackets, or PCGE will not recognise that a mnemonic is going to be entered. Operators are introduced using next keys:

| Operator | Key |
|---|---|

[2] These addresing mode has been taken from the Scanner41 memory map when PCGE was designed and has been mantained since then.

| | |
|---|---|
| AND | & |
| NOT | ! |
| OR | \| |
| XOR | ^* |

## 3.6 Graphical edition

PCGE graphical features simplify user operation. They provide important cost savings when a GPC design becomes complicated. In this way, PCGE is a useful and flexible tool. Graphical edition features are mainly located in *Edit* and *Show* menus.

Undo/redo operation is also available. If one insertion is deleted with their bit comments programmed, undo operation can recover previous state. On the contrary, if it is an entity what is deleted, undo operation recovers the entity with the default LT, not the original one.

Next points detail graphical features of PCGE.

### 3.6.1 Copy/Paste

There are several ways to make this operation, but all produce the same result: copy/paste insertion, entities or labels from existing.

This operation is divided in to two steps; first one is copying desired elements onto the clipboard. The other is pasting elements from the clipboard. Since user can do operations between copy and paste actions, some checks and cautions must be regarded before pasting, or the GPC file can become corrupted.

Paste operation can be realised in the same GPC or between GPC's. It also can be done between blocks or in the same block.

Restrictions:

- ***Design***

The following operations made before pasting invalidate the operation:

- If block is edited so that resizable entities are forced to non-resizable.
- Entities deletion in block.
- Modification of the LT's set associated to the insertion's behaviour.
- Modification of the colour set.

If insertions have *Text* or *Attribute Text* entities:

- Font's set modification.

If copied elements are *labels*:

- Modification of labels set, addition of a new label type.

- Modification of copied label type.

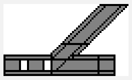
When copying between two GPC's, it must be taken into account that both files must have same library, set of labels, logic tables, colours and fonts.

Copy is made in the same origin point than copied element, therefore elements pasted in the same page will be pasted over the original.

If in the design page where insertions are pasted insertions with the same name as that copied already exist, PCGE automatically renames pasted insertions with the predefined auto-name and issues a message reading which insertions were renamed.

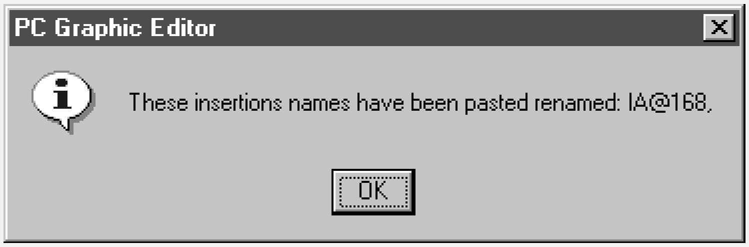
Example 3.6 — 1: Copy/paste one insertion in the same page.

Insertion to be copied.



Insertion's name: IA@168

It is copied and pasted in the same page, this way a message advising that insertion IA@168 has been renamed is issued.



These insertions names have been pasted renamed: IA@168,

Once accepted the message, two identical insertions are obtained.

Bit comment configuration is copied and pasted along with graphical properties.

- ***Blocks***

When copy/paste operation is done between button and non button blocks, origin behaviours are not pasted in target.

### 3.6.2 Duplicate

Duplicate operation has the same effect than copy/paste but is faster. It is done in the same page and in the same GPC. Duplicate is also available in blocks entities.

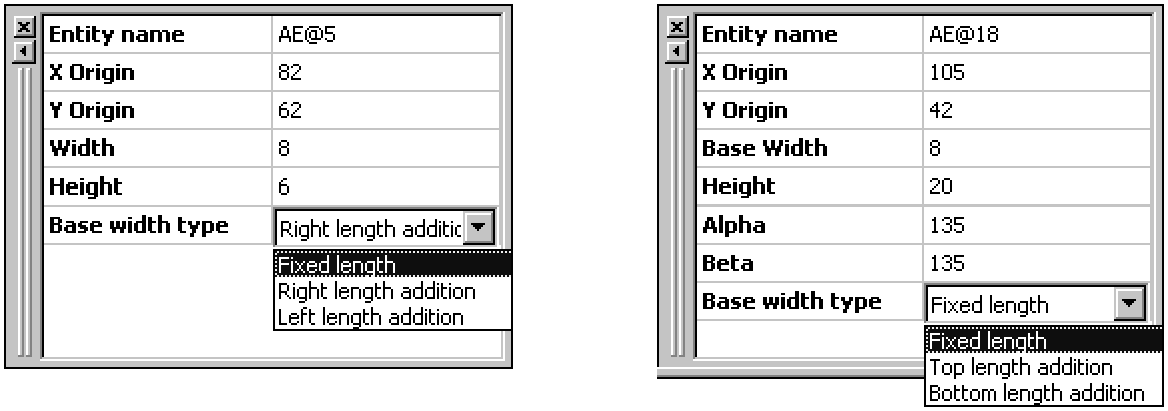
Duplicated elements are auto-named by PCGE.

### 3.6.3 Resize

This is a special operation for *FillRect* and *Parallel* entities. The permitted resizing type is defined at entity level, but resizing is made at insertion level.

FillRect entities can be resized to the left or to the right. Parallel entities can be resized top and down.

This property can be found along with the other characteristics of the entity.



| | |
|---|---|
| **Entity name** | AE@5 |
| **X Origin** | 82 |
| **Y Origin** | 62 |
| **Width** | 8 |
| **Height** | 6 |
| **Base width type** | Right length additic |

Fixed length
Right length addition
Left length addition

| | |
|---|---|
| **Entity name** | AE@18 |
| **X Origin** | 105 |
| **Y Origin** | 42 |
| **Base Width** | 8 |
| **Height** | 20 |
| **Alpha** | 135 |
| **Beta** | 135 |
| **Base width type** | Fixed length |

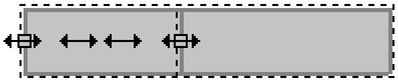
Fixed length
Top length addition
Bottom length addition

**Figure 3. 21 — Fill Rect's properties and Paralel's properties**

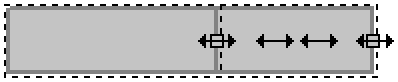
One resizable insertion will show a handle for resizing when selected. Handler will be visible in the resizing side. Mouse cursor changes its appearance when is passed over the handler.

Mouse cursor appearance is ⟷ over *Fillrect* resizing handlers and ↕ for parallel.

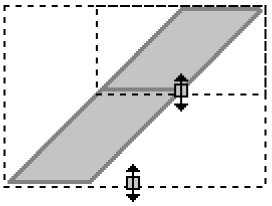
To resize an insertion, place mouse over the handler and drag. PCGE allow fine adjustment combining muse position and use of keyboard cursor keys. This way, resizing pixel-by-pixel can be obtained.



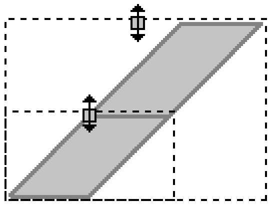
**Right length addition**



**Left length addition**



**Bottom length addition**



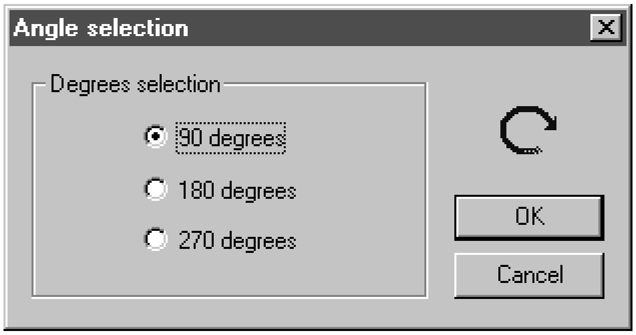
**Top length addition**

### 3.6.4 Rotate

Insertions can be rotated except those containing resizable entities or bitmap entities. Blocks and labels cannot be rotated.

Only positive angles are permitted. Negative angles are transformed to supplementary angles (i.e. –90º to 270º). Only 90º, 180º and 270º are permitted.

Rotation can be applied directly by the *Angle* property in the property view. Other way is using toolbar *Rotate.* Finally, menu option *Edit/Rotate*…(Ctrl+R) displays the next dialog.



Angle selection

Degrees selection

● 90 degrees

○ 180 degrees

○ 270 degrees

**Figure 3. 22 — Angel Selection dialog**

The figure 3.23 shows the result of rotating a fillrect and a text.

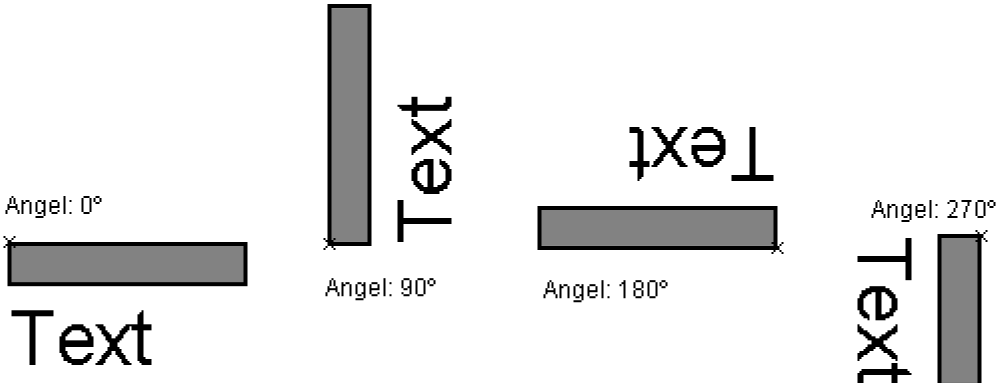


**Figure 3. 23 — Rotated Insertions**

### 3.6.5 Zoom

Let the user work in different sizes. Scale factor is stamped in the left side of status toolbar.

Zoom normal is fixed to 100% in design view and 400% in block view. Zoom normal can be restored with key shortcut Ctrl+L, or by means of menu *Show/Zoom normal*.

**Zoom to Fit** button, placed in *Zoom* toolbar calculates a zoom factor so that all elements just fox in the view area.

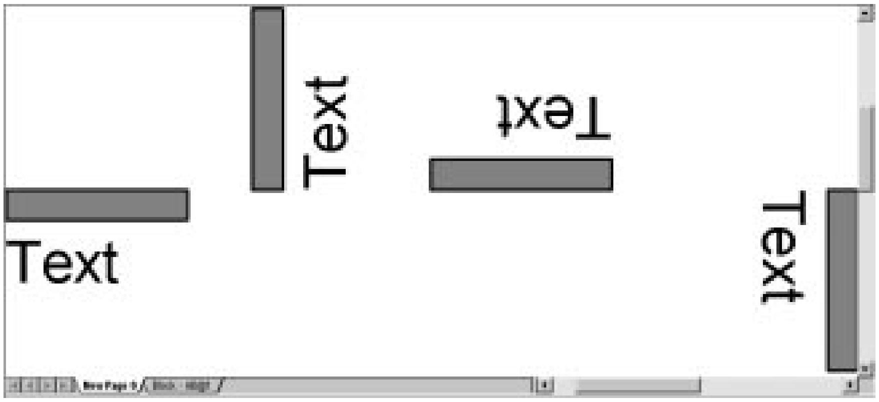


**Figure 3. 24 — Zoom to Fit**

**Zoom to Selection** button calculates the necessary zoom factor to get selected elements closest.

Following the example of figure 3.24, selecting the non rotated insertion and click on **Zoom to Selection** button would produce the result shown by the figure.

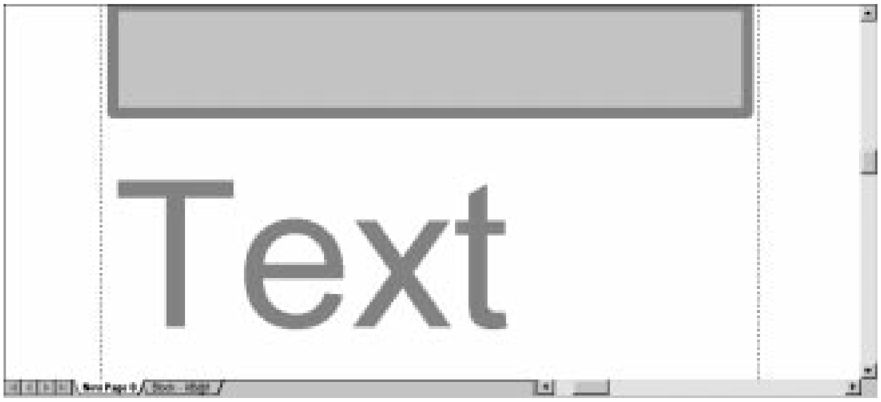


**Figure 3. 25 — Zoom to Selection**

Zoom button in Zoom toolbar enters in zoom mode. In zoom mode, left button click produces a zoom factor x2 and right mouse button by /2.

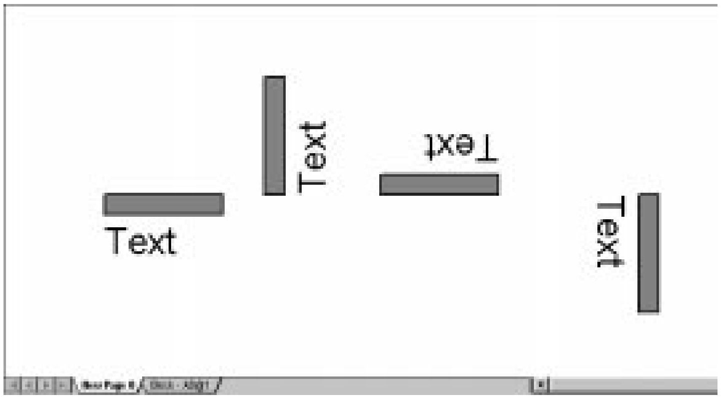


**Figure 3. 26 — Zoom 200 %**

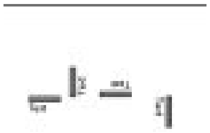
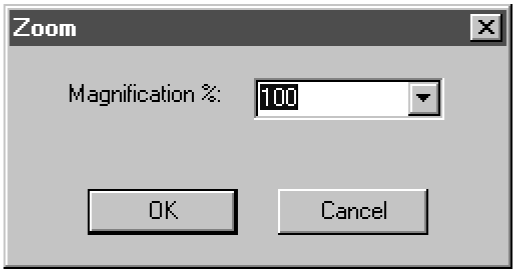


**Figure 3. 27 — Zoom 50 %**

Predefined zoom factors as 50%, 75%, 100% and 200% are available in PCGE in menu option *Show/Zoom Percent*.

PCGE also allow the possibility of generating any zoom factor with option *Show/Zoom Custom*… This option displays a dialog to enter the desired zoom factor.



Magnification %: 100

**Figure 3. 28 — Zoom Custom dialog**

Finally, figure 3.29 show the aspect of a design and block edition view with zoom normal in both cases.

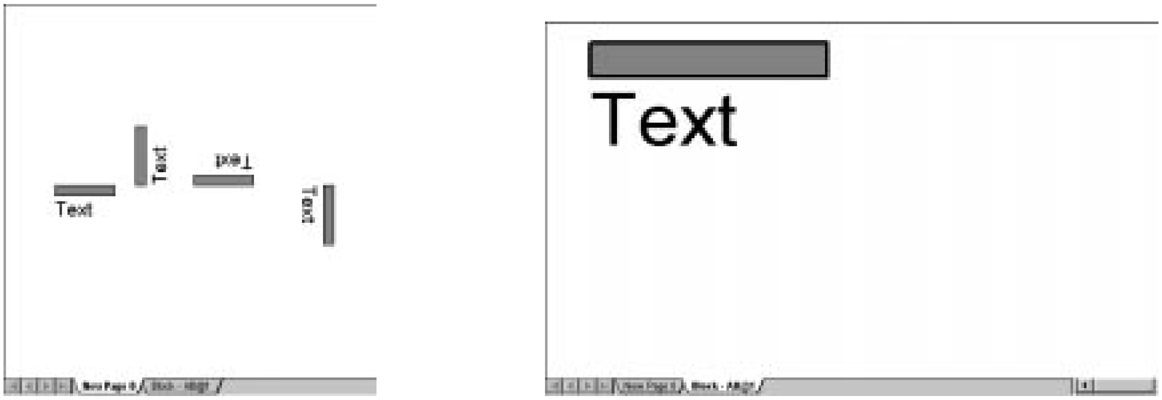


**Figure 3. 29 — Zoom Normal in a Design and in a Block**

### 3.6.6 Multiple selection

PCGE supports multiple selection..

Multiple selection can be done in two ways: selecting elements individually or some of them maintaining at the same time. To use the first option, just select one element and the rest of elements while maintaining *Shift* or *Ctrl* key pressed. The second one can de done using a capture window, i.e., dragging mouse to outline a shape containing all the elements to be selected.

Menu option *Edit/Select All* (Ctrl + A) select all elements in the active view (insertion/labels or entities).

### 3.6.7 Alignment

PCGE supports alignment of elements in design and block view.

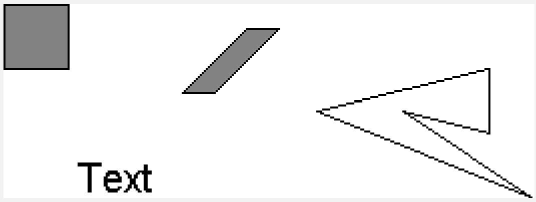
Alignment will produce alignment of the previously selected elements, taking as reference the last one selected.

Alignment types available are:

- Top.- To align elements according to the upper part of the reference element.
- Middle.- To align elements according to the imaginary central line of the reference element.
- Bottom.- To align elements according to the lower part of the reference element.
- Left.- To align elements according to the left side of the selected element.
- Center.- To align elements according to an imaginary vertical axis of the selected element.

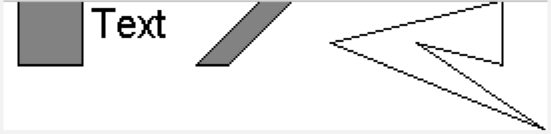
Example 3.6 — 2: Alignment

Suppose a set of blocks composed by entities FillRect, Line, Text, FillCircle, Parallel and Polygon. We have insertion of that block distributed in the design view too.

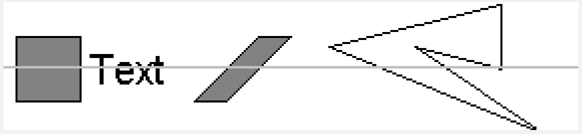


All alignment will be realised against the polygon (last selected insertion).

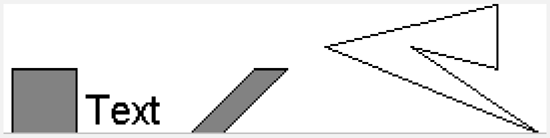
a) Top



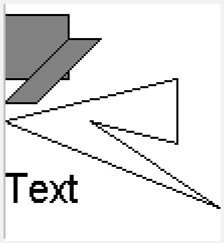
b) Middle



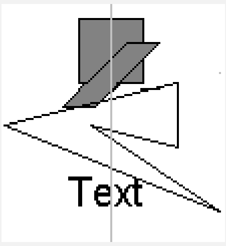
c) Bottom



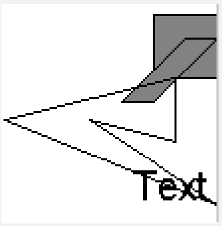
d) Left
.



e) Center



f) Right



## 3.7 Settings of the design.

### 3.7.1 Blank Char

BlankChar is a important parameter of the design. This is the character used in GPC file to define the blank gaps. The current BlankChar of a design cannot be used in name of insertions, labels, blocks, Logic Tables, entities, behaviours, etc.

BlankChar must be selected from the list of permitted blankchars:

! “ # $ & ‘ * + . / : ; < > ? \ ^ _ ` { | } ~ %

BlankChar selection is made through a dialog after selecting *File/Select BlankChar*…

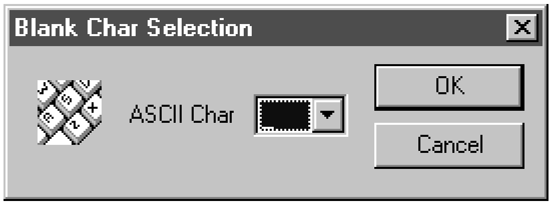
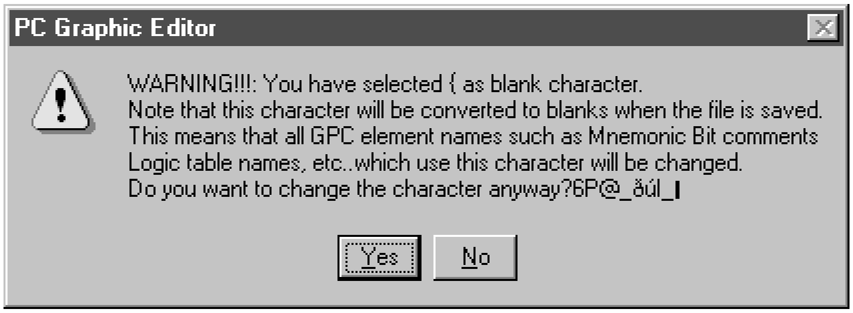


**Figure 3. 30 — Blank Char Selection dialog**

Once selected the blankchar, a warning is issued telling about this operation change.



WARNING!!!: You have selected { as blank character.
Note that this character will be converted to blanks when the file is saved.
This means that all GPC element names such as Mnemonic Bit comments
Logic table names, etc..which use this character will be changed.
Do you want to change the character anyway?6P@_ǎúl_▌

**Figure 3. 31 — Blank Char Warning**

When a GPC file is created, a default BlankChar (‘_’) is selected. Should this blankchar were not suitable for your design, first thing to do is changing the blankchar.

Apart from blankchar, GPC elements also have a list of forbidden names, summarised in the table below.

| GPC Element | Forbidden characters in name. |
|---|---|
| Mnemonics | BlankChar, ‘[‘ and ‘]’ |
| Bit comments | CLT: ‘;’, BlankChar<br>ELT: ‘;’, ‘[‘, ‘]’, BlankChar |
| Insertions | BlankChar, ‘[‘, ‘=’, ‘]’ |
| Blocks | BlankChar, ‘[‘, ‘=’, ‘]’ |
| Entities | BlankChar |
| Pages | BlankChar |
| Labels | BlankChar |

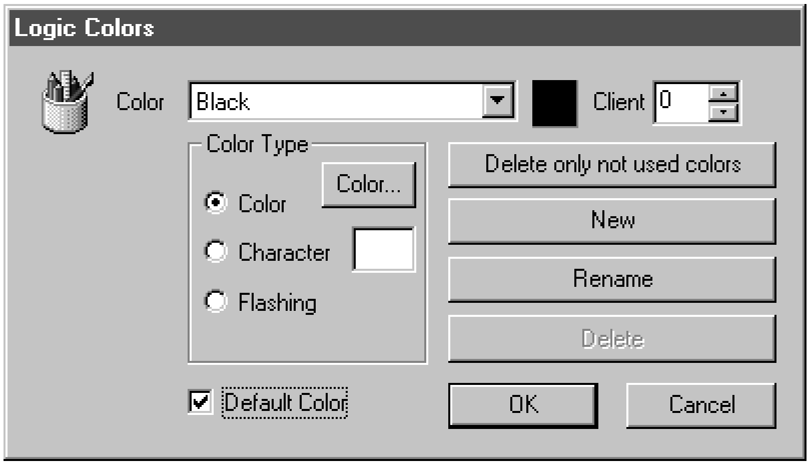
### 3.7.2 Colors

Logic Tables and Labels use colours.

PCGE manages three different type of colours:

- Color.- Is selected from the available colour palette.
- Character.- A colour is defined using a character. Those colours are processed in a special way in client systems when recognised.
- Flashing.-Are colours taken from the palette, but displayed in a flash way.

A predefined set of colours are created when a new design is created (*See 3.1 Creating a new design*). Only not used colours can be deleted. Colours configuration is done using the dialog trough menu *option Options/Edit Colours*…



Logic Colors

Color: Black ■ Client: 0

Color Type

● Color

○ Character

○ Flashing

☑ Default Color

**Figure 3. 32 — Logic Colors dialog**

***Color***

List of colours available in the graphic editor at this moment. A sample of the colour is shown to the right.

***Client***

Client Index of the colour. The user can change this index but it must be unique among all colour indexes.

***Color Type***

Type of the colour (colour, character and flashing). If colour is character, and edit box permits character introduction. **Color** button displays windows colour palette to select the desired colour.

***Default Color***

Check box to indicate that is the default colour of the design. This colour will be assigned by default to elements when they are created. This is applied to labels and logic tables. Default colour is unique, so if one colour is marked as default, the previous default colour stops being selected. When a new GPC is created, black is the default colour.

***Delete only not used color***

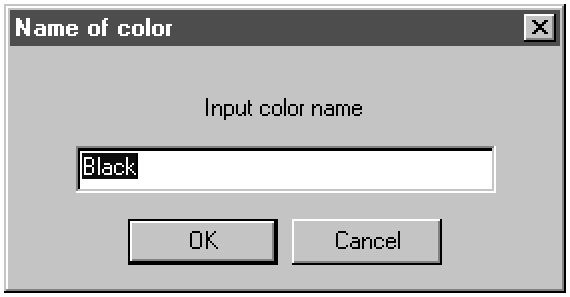
This button deletes colours not used in logic tables nor labels.

***New***

Aggregates a new colour to the existing set of colours. A dialog to enter the new colour name is displayed.

***Rename***

Uses the same dialog as *New* to change the name of a colour.



Black

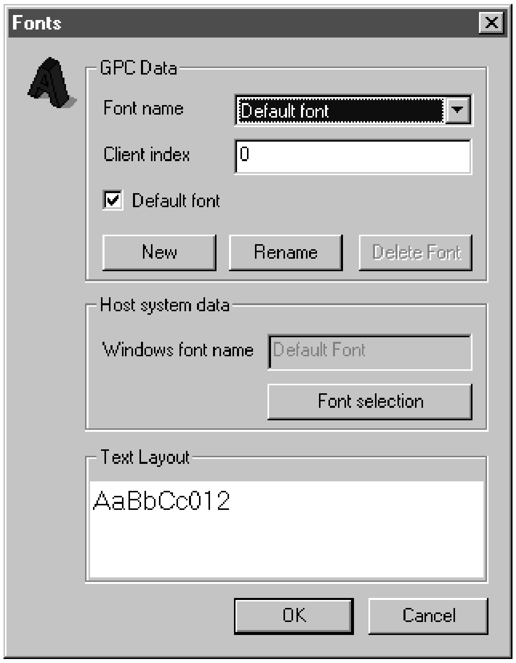
**Figure 3. 33 — Name of Color (new — rename)**

***Delete***

Delete the current selected colour. This button is active only is colour is not being in use.

### 3.7.3 Fonts

Text and Attribute Text entities and labels use fonts. When a new GPC design is created a set of fonts is created too *(See 3.1 Creating a new design)*. Fonts are editable by user at any time. To configure fonts select menu option *Options/Edit Fonts*…A font edition dialog is displayed.



Fonts

GPC Data

Font name: Default font

Client index: 0

☑ Default font

Host system data

Windows font name: Default Font

Text Layout

AaBbCc012

**Figure 3. 34 — Fonts dialog**

We can see three areas in the dialog. In the upper area it is located information of about how the font addresses are in the GPC file, in the middle it appears how fonts are displayed depending on the computer in which PCGE is running. Finally, the lower area shows an image of how font is displayed.

### *Font Name*

This list is used to select a GPC font.

### *Client Index*

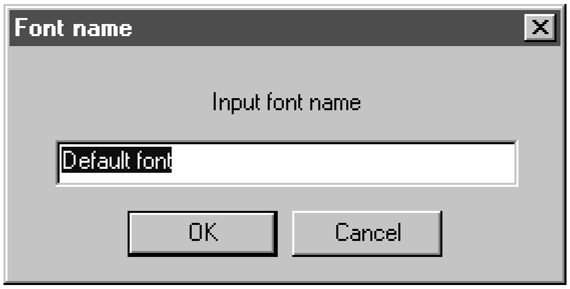
Client index of the GPC font. Font client indexes are used by non-Windows client systems. The user can edit those indexes.

### *Default Font*

It marks the default font. As in colours, a default font must exist in a GPC design so that a new label, text or attribute text can have associated GPC font when created. A GPC font can be set as default at any time by, deselecting the previous default font.

### *New*

Creates a new font, user configurable. Font name introduction is done using a dialog.



Font name

Input font name

Default font

**Figure 3. 35 — Font name (new — rename)**

***Rename***

User the same dialog as in *New*. Renames a font

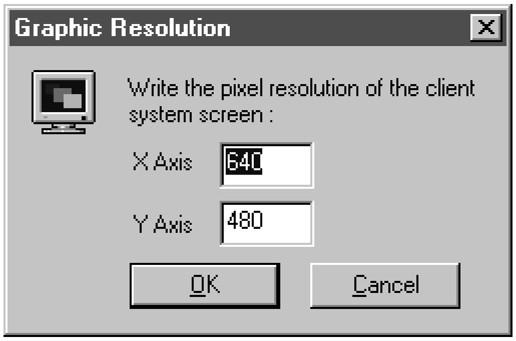
***Font Selection***

This option displays the standard windows dialog of font selection.

### 3.7.4 Resolution

Resolution is a property that affects the design view. It is the size in pixels of every page, and does not have anything in common with computer resolution.

This property can be configured through menu option *Options/Resolution…*



Graphic Resolution

Write the pixel resolution of the client system screen :

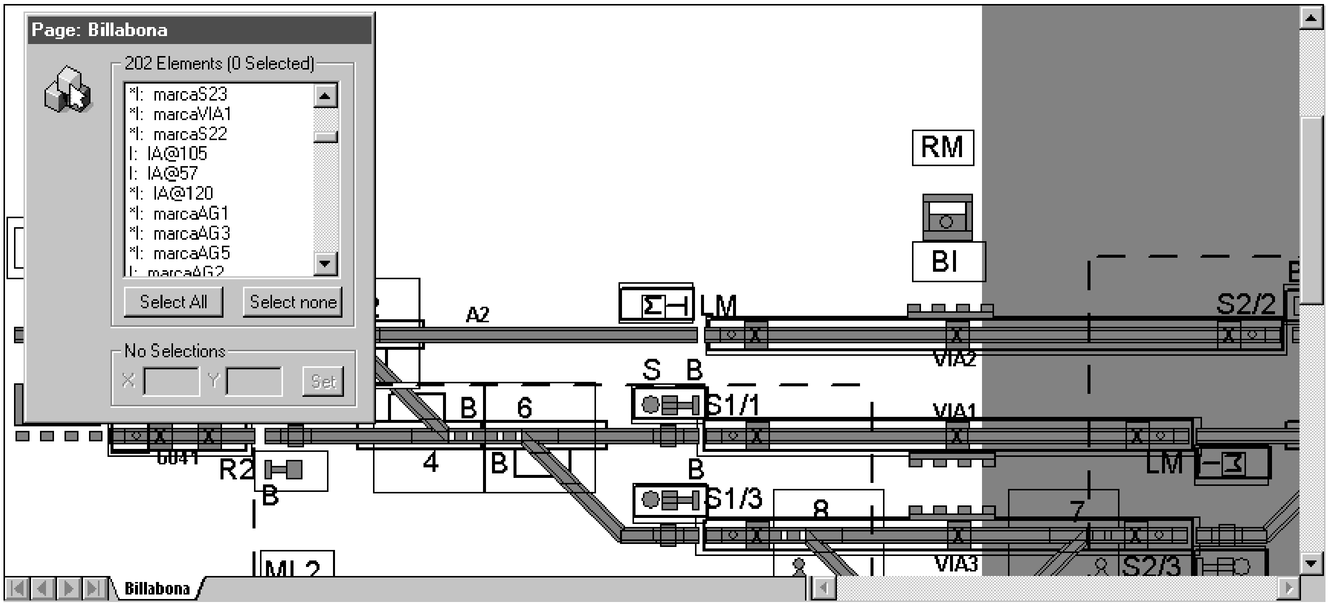
X Axis 640

Y Axis 480

**Figure 3. 36 — Graphic Resolution dialog**

Reducing design resolution may cause some insertion/labels lay outside the design area. Although they continue being displayed, no edition action can be done on them.

If this happens, insertion/labels that lay outside the design area will be marked with asterisk (‘*’) in the dynamic list (*Show/Show Dynamic List*). See figure 3.37 to observe this undesirable effect.



Page: Billabona

202 Elements (0 Selected)

- *I: marcaS23
- *I: marcaVIA1
- *I: marcaS22
- I: IA@105
- I: IA@57
- *I: IA@120
- *I: marcaAG1
- *I: marcaAG3
- *I: marcaAG5
- I: marcaAG2

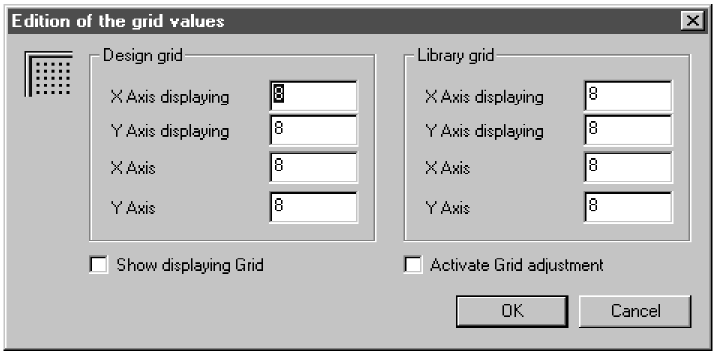
No Selections

X Y

Figure 3. 37 — Graphic Resolution less than necessary in a design

### 3.7.5 Grid

Grid aids the user to use references when drawing. Grid values can be configured. Default value is 8 pixels.



Edition of the grid values

| Design grid | | Library grid | |
|---|---|---|---|
| X Axis displaying | 8 | X Axis displaying | 8 |
| Y Axis displaying | 8 | Y Axis displaying | 8 |
| X Axis | 8 | X Axis | 8 |
| Y Axis | 8 | Y Axis | 8 |

□ Show displaying Grid

□ Activate Grid adjustment

Figure 3. 38 — Grid dialog

Menu option *Options/Edit grid*… displays grid dialog. Note that grid for library view is also available.

*Show displaying Grid* shows/hide grid in the same way than it does correspondent button in *Canvas* toolbar. *Activate Grid adjustment* provokes snap to grid mode. There is a button to do the same in *Canvas* toolbar either.

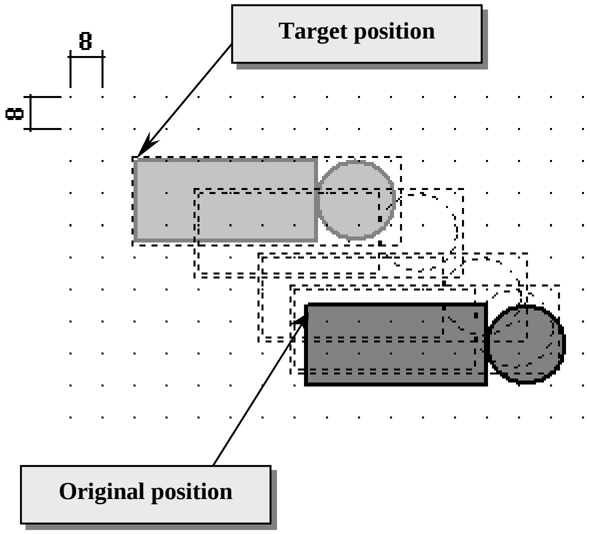


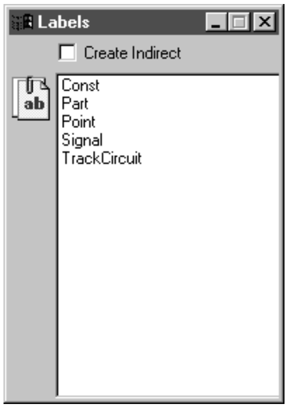
**Figure 3. 39 — Insertion moved**

### 3.7.6 Labels and labels type

Labels are descriptive text, used in design view. Labels do not use logic tables associated, but they use fixed colours. There are two kind of labels: direct and indirect.

Select menu option *Show/Labels Toolbar* to obtain the toolbar of labels. This toolbar is a floating and resizable toolbar, and it has a list of preconfigured direct label types existing in a GPC design plus a checkbox to introduce an indirect label.

Once label toolbar is displayed, just select desired label type from the toolbar by click on them and click over design view to insert a new label of that type.



Const
Part
Point
Signal
TrackCircuit

**Figure 3. 40 — Labels Toolbar**

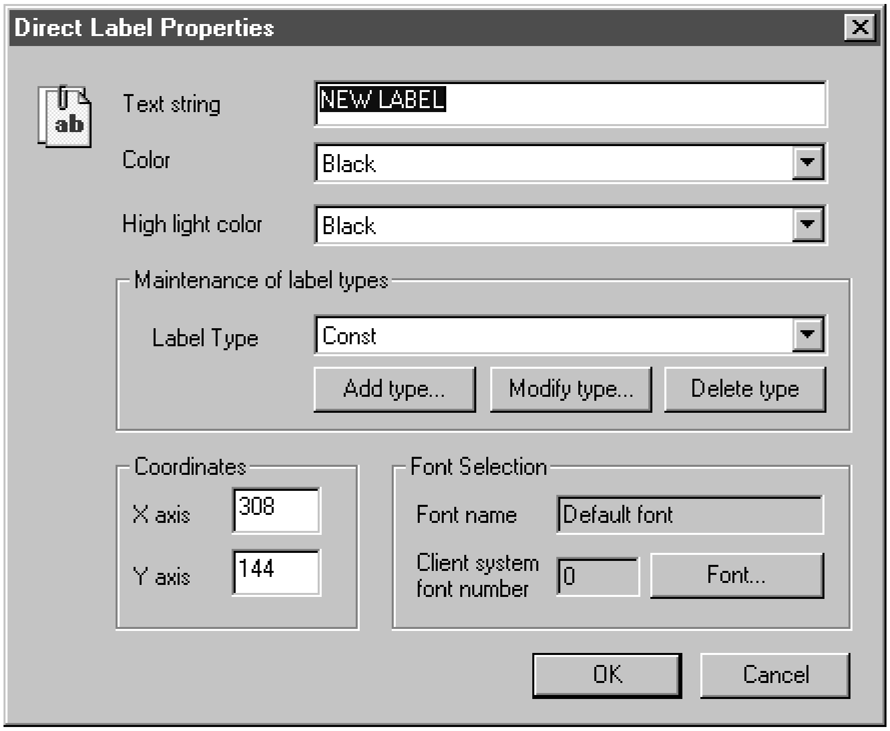
Normally most of labels used in a design are direct. Direct labels are used to group labels for types as signals, points, track circuits, etc. Predefined types are:

- Const

- Part
- Point
- Signal
- Track Circuit

User can create his new types or modify any of the predefined.

Double-click on a label in the design will display its label properties dialog.



Text string: NEW LABEL
Color: Black
High light color: Black
Label Type: Const
X axis: 308
Y axis: 144
Font name: Default font
Client system font number: 0

**Figure 3. 41 — Direct Label Properties dialog**

### *Text String*

Text of the label.

### *Color*

Colour of the label. This is the label's colour in normal state.

### *High Light Color*

This is the label's colour if the label is associated to a transparent button and the button is selected. This property can be used by client systems as for example SCC.

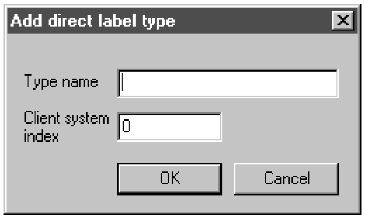
If this feature is used then is highly recommendable that colours, normal and highlight be different.

### *Label Type*

List with available label types. The selected type is the one which label was created. Using this control label type can be changed.

### *Add Type…*

Adds a new label type. A label's type consist on a name and a client index. The index must be unique.



Type name

Client system index 0

**Figure 3. 42 — Direct Label Properties dialog**

### *Modify Type*

This button allows user to edit a label type. The dialog is the same as previous.

### *Delete Type*

Delete a label type if it is being unused by any label.

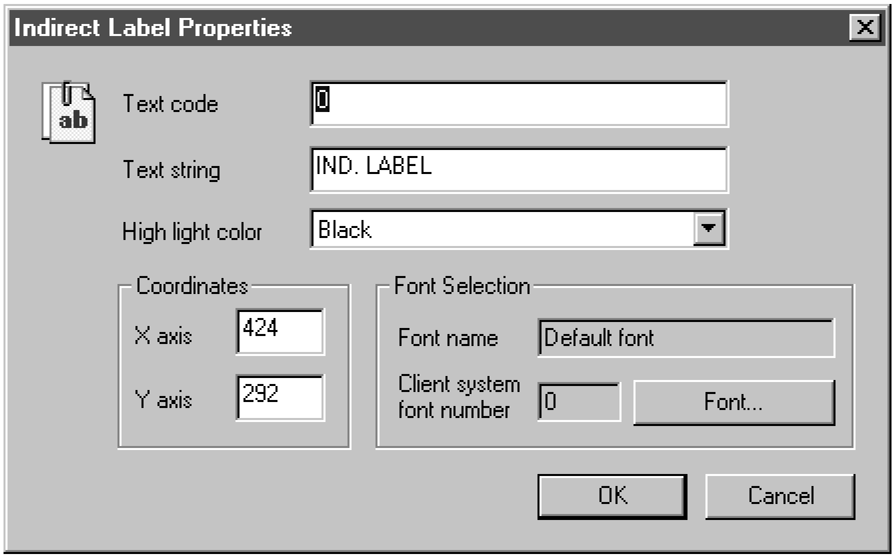
### *Coordinates*

Change label location co-ordinates in the design view. Co-ordinates are referred to the left-bottom point of the label.

### *Font Selection*

Those controls are used to change the system font associated to the label, so it displays the explained dialog of fonts.

Indirect labels are of general use. A bit code is alloted to them so that client systems can perform actions on them. Double-click on one indirect label will display its properties dialogue.



Text code 0

Text string IND. LABEL

High light color Black

X axis 424

Y axis 292

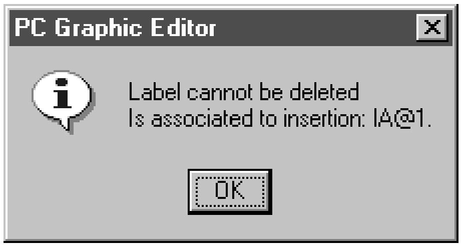
Font name Default font

Client system font number 0

**Figure 3. 43 — Indirect Label Properties dialog**

This dialog is quite similar to the one for direct labels. Text code is the only difference.

Labels can be deleted except if they are associated to a button. In this case, a warning is issued. If label is associated to more than one button more than one warning are issued.



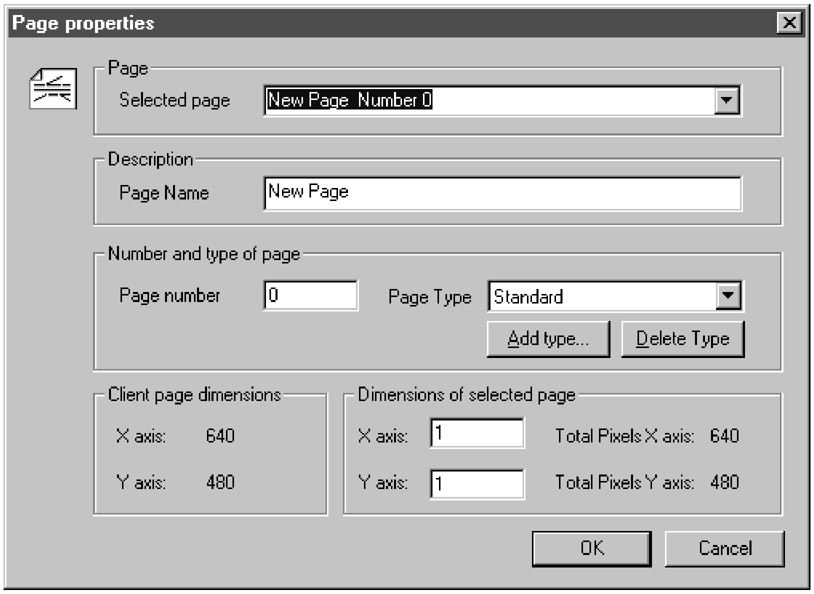
PC Graphic Editor

Label cannot be deleted
Is associated to insertion: IA@1.

**Figure 3. 44 — Label associated to button insertion**

### 3.7.7 Page Properties

Page properties dialog groups all relevant page properties in a design view.



Page properties

Page
Selected page: New Page Number 0

Description
Page Name: New Page

Number and type of page
Page number: 0
Page Type: Standard

Client page dimensions
X axis: 640
Y axis: 480

Dimensions of selected page
X axis: 1
Total Pixels X axis: 640
Y axis: 1
Total Pixels Y axis: 480

**Figure 3. 45 — Page properties dialog.**

Information is grouped in 5 areas. At the top it appears the name and number of page. It consists of a list with all the page in the design. The list allows moving across the rest of pages and edits several pages without exiting the dialogue.

Just below it is an edit box to edit the page name.

In the middle the page number and page type are grouped. Different characteristics can be assigned to different page types, for example, resolution. In order to create a new page type click on **Add Type…** button and fill the corresponding dialogue. Default page type is Standard.

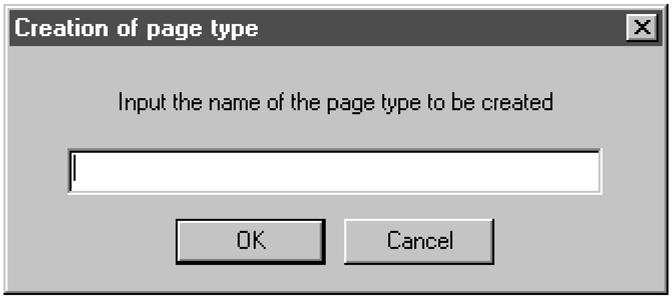


**Figure 3. 46 — New Page Type dialog**

**Delete Type** button deletes an existing page type previous confirmation to user.

Left bottom area informs about size of client page. Right bottom area is used to multiply by an integer factor page resolution.

### 3.7.8 Compress GPC file

This option is located in *File* menu, and allow user to save GPC file in a compress or expanded format. By default is compressed. If compressed, all items in GPC format will be saved compressed, and as a result, GPC file will occupy less storage size.

# 4 Installation and setup

This chapter describes how to set up PCGE on a PC.

## 4.1 Minimum Platform Requirements

PCGE requires an IBM-compatible 350 MHz Pentium II PC with the following specifications for a reliable and timely operation:

- 64 Mb RAM.
- Graphic Resolution: 800x600x256c
- 20 Mb free hard disk.
- Windows NT 4.0 with service pack 5 or 6 installed.

## 4.2 Setup Process

The PCGE tool shall execute in a PC environment, running Windows 2000, since it had been developed in it.

The PCGE tool will show a warning when executing in Windows NT 4.0 service pack 4 or Windows 95/98 because PCGE will not be specifically tested under those environments.

To install PCGE on your computer, run PCGE_NT_SETUP.EXE file and follow all steps provided by the installation program.

# 5 Compatibility

Next it is presented a table which summarises all PCGE version along with GPC format file version generated and read. This table can be used for compatibility between versions.

| | **Read** | **Write** |
|---|---|---|
| **PCGE version** | **(GPC version)** | **(GPC version)** |
| P04, P06 | 1.03 | 1.03 |
| P08, P09 | 1.03<br>1.04 | 1.04 |
| 1.0, 1.1 | 1.04<br>1.05 | 1.05 |
| P12, P15 | 1.05<br>1.06 | 1.06 |
| P16, P18 | 1.06<br>1.07 | 1.07 |
| 2.0, 2.1, 2.2 | 1.05<br>1.08 | 1.08 |
| 3.0, 3.01, 3.02, 3.03 | 1.07<br>1.08<br>2.00 | 2.00 |
| 3.1 | 2.00 | 2.00 |

# 6 Other tools

## 6.1 GPCLibraryManager.

GPCLibraryManager manages block library elements between two GPC files. Using GPCLibraryManeger allow import/export blocks in a safe way, maintaining GPC format integrity.

The figure 6.1shows the GPC Library Manager User Interface

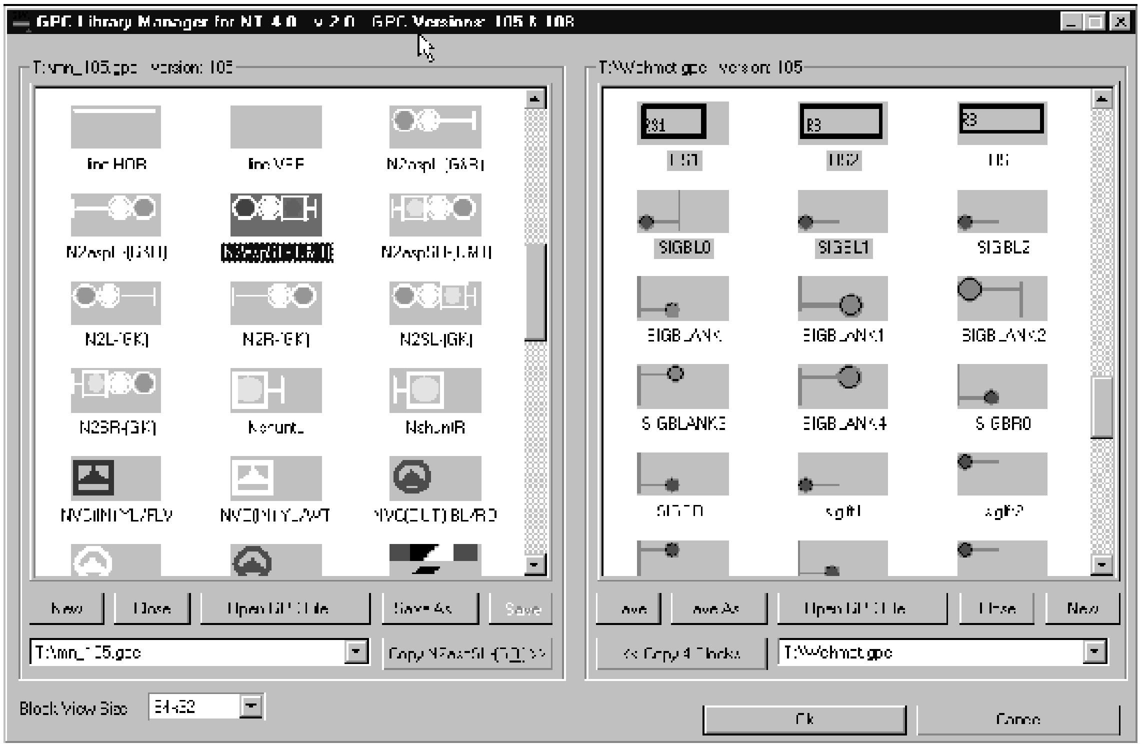


**Figure 6. 1 GPCLibraryManager main window.**

The application has 2 list controls each of them containing the blocks read from a GPC file. The way to manage these controls is similar to that of the right pane of the windows explorer. The operations you can carry out are:

- Select one block by clicking on it.
- Select a group of blocks using Control, Shift and arrow keys.
- Select a group of blocks dragging the mouse at the same time that the left button is pressed.
- Rename a block by clicking on its name

Find a block starting with a letter, by selecting a block and then pressing the letter in keyboard. The next block starting with the letter will be found.

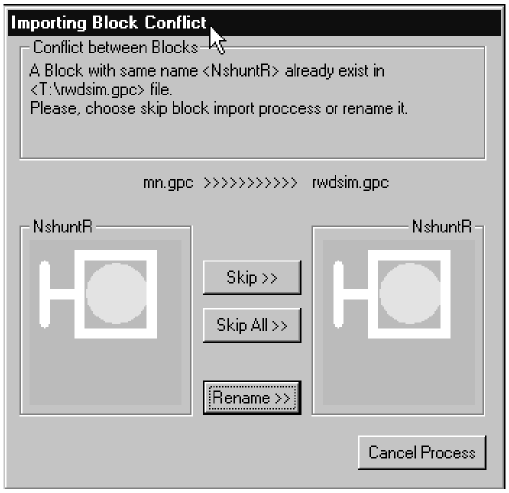
Invoke a pop-up menu with the right button to perform the most common actions, like copy or rename block.

### *Dialog Fields Description:*

| **Field Name** | **Description** | **How to use** |
|---|---|---|
| New | Create a new empty GPC file | This button is enabled only when another file is opened in the opposite window. The version of the new file created is the same as the other file opened. |
| *Close* | Close current GPC file | Click button. The List control Window will be disabled. |
| Open GPC File… | Open a GPC file. | Click button. Windows Open dialog will be shown to select a file. Note that only versions 1.05 and 1.08 of GPC files are supported. |
| Save As… | Save current opened GPC file. | Click button. Windows Save dialog will be shown to select a target file. |
| Save | Save current opened file with the same name. | Click button. The changes done in file will be saved to it. |
| Copy Block | Copy the block/blocks selected to the opposite GPC file. | Click button. The blocks will be copied from source GPC file to the target one. |
| Historic of Opened Files | Hold all the file names opened in a session | Click on combo box and select a GPC file name. It will be opened and its blocks will be shown in a List Control |
| Block View Size | This control allows to select the size of the block icons | Click on combo box and select a size. All the block icons will be resized to its new size. |

### *Conflict Dialogs:*

The PCGE is not able to manage elements of the same type with repeated names (for example two different logic tables but with the same name). This means that when a element like a colour a logic tables, etc.. in the source file has the same name as the one defined in the target file, and they are different a conflict dialog will be shown to solve the problem. The aspect of those dialogs is shown in figures 6.2 and 6.3.



Importing Block Conflict

Conflict between Blocks

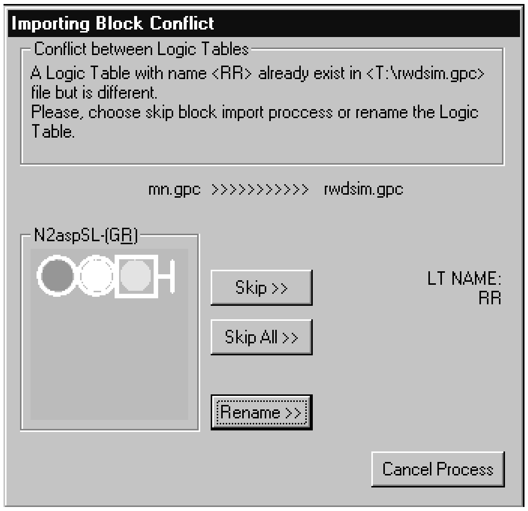
A Block with same name <NshuntR> already exist in <T:\rwdsim.gpc> file.
Please, choose skip block import proccess or rename it.

mn.gpc >>>>>>>>>>> rwdsim.gpc

**Figure 6. 2 Conflict dialog**

If a block conflict is found a dialog will be shown. The possibilities offered are: skip the block copy, skip current and following blocks with the same problem or copy source block renamed.

Also the possibility of cancelling the process is provided.



Importing Block Conflict

Conflict between Logic Tables

A Logic Table with name <RR> already exist in <T:\rwdsim.gpc> file but is different.
Please, choose skip block import proccess or rename the Logic Table.

mn.gpc >>>>>>>>>>> rwdsim.gpc

LT NAME:
RR

**Figure 6. 3 Comflict when importing block dialog.**

If a logic table, font or colour conflict is found, a dialog like the one shown above will be displayed. The possibilities offered are skip the block copy, skip current and following blocks with the same problem or copy source logic table, colour or font renamed.

Also the possibility of cancelling the process is provided.

## 6.2 DtMC

Before MAA, bit comments programming was done using direct bits. Those bits support programming using Line, Housing, Card and Bit format. When the GPC design is made for a client system which is going to be connected to a Westrace a mechanism must be provided to make a correspondence between direct bits and mnemonics used by Westrace system, or to use mnemonics directly. Moviolaw 2.0 for NT only accepts mnemonics, consequently an application which converts old direct bits programmed GPC files using L.H,C,B format into GPC files with mnemonics is needed.

Mnemonic file is named file.MGR, and it contains a list of mnemonics that uses GPC file from an external system (i.e. Westrace). Assuming that L,H,C,B parameters are programmed starting in 1,0,0,0; and that each card has 32 bits, and that each housing has 16 cars, the conversion is done in the next way:

| **MGR file** | **Associated L,H,C,B.** |
|---|---|
| *Mnemonic 1* | 1,0,0,0 |
| *Mnemonic 2* | 1,0,0,1 |
| *…* | |
| *Mnemonic 32* | 1,0,0,31 |
| *Mnemonic 33* | 1,0,1,0 |
| *…* | |
| *Mnemonic 512* | 1,0,15,31 |
| *Mnemonic 513* | 1,1,0,0 |
| *…* | |
| *Mnemonic n* | 1,(int) [(n-1)/512],(int)[mantissa of H]*16l,(int)[mantissa of C]*32<br>A possible algorithm used to express this into C++ could be:<br>1, (H=(n & 0xFE00)>>9),(C=(n & 0x01E0)>>5),(B=(n & 0x001F)-1) |

***Note***

The mathematical term mantissa is strictly referred to logarithms, but here is used because the idea is the same: the part of the number in the right of the decimal point.

Here we are a concrete example of this algorithm:

Imagine the necessity to translate to L,H,C,B format an MGR mnemonic in the 1000 position in MGR file.

Line is always 1 **(L = 1)**

To calculate the housing, divide 999 (n-1) by 512. 512 is the result of multiply 16 cards by 32 bits, and it is the total number of bits contained in a single Housing.

999/512 = 1,951171875. The entire part of this number is 1, so it is the Housing. **(H=1)**

To calculate the card we use the mantissa of the previous number.

Mantissa = 0,951171875

Multiplying by 16 obtains the entire part.

0,951171875* 16 = 15,21875 => entire part = 15. Card = 15 **(C=15)**

Finally, to obtain the bit it is taken the entire part of the last's number mantissa multiply by 32

Mantissa of 15, 218755 => 0, 21875

0, 21875 * 32 = 7 => bit = 7 **(B=7)**

1000 = 1,1,15,7 in format L,H,C,B.

Once explained the conversion, the application must do the following:

To read a GPC file containing bit comments programmed in L,H,C,B format.

To read an MGR file.

Convert all direct bits to mnemonics.

Associate the direct bits bit numbering to mnemonics from MGR file using the reverse algorithm of the one explained above.

**N = H*512 + C*32 + B + 1**

Number N calculated in the formula above is the N entry to the MGR file. It is the index to obtain the correspondent mnemonics from the MGR file.

Program the correspondent mnemonic in the mnemonic field of the insertion.

DtMC is the tool designed to do the previous A-F points. When invoked it is shown a dialog as:

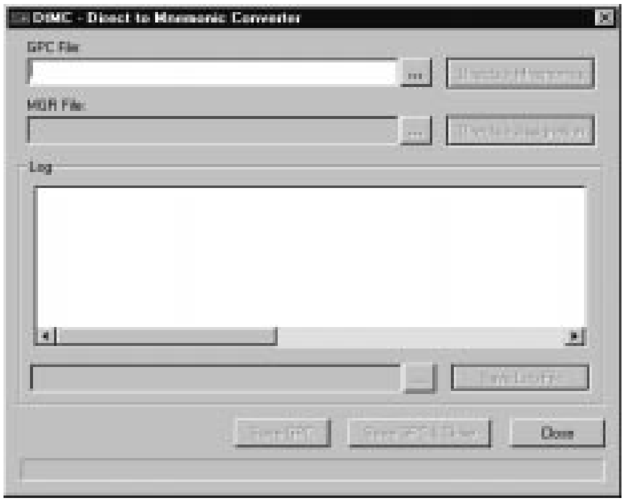
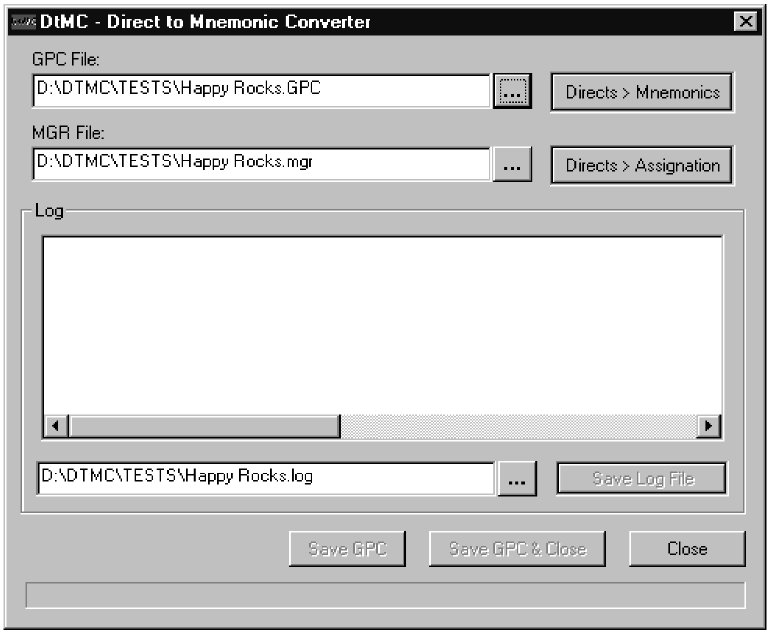


**Figure 6. 4 DtMC dialog.**

The first step is opening the GPC file which direct bits must be passed to mnemonics. To do that, click on navigation button in edit box of *GPC* file. When GPC file is open note that DtMC tries to locate a MGR file with the same name in the current directory. It also proposes a default name and location of the log file generated with the result of the conversion.



GPC File: D:\DTMC\TESTS\Happy Rocks.GPC

MGR File: D:\DTMC\TESTS\Happy Rocks.mgr

D:\DTMC\TESTS\Happy Rocks.log

**Figure 6. 5 Data loaded in DtMC**

At this point, user may want to perform one of two operations: only convert direct bits to mnemonic so that they can be assigned using MAA later, or convert direct bits to mnemonics and automatically assign them if L, H, C, B information is already programmed.

In the first case, click button ***Direct > Mnemonics***. Then, conversion will be done. When converting from direct to mnemonic, DtMC expects that all bit comments are originally set as direct bits, if it find bit comments without direct format, a warning advice is issued, but conversion is not interrumped in any case.

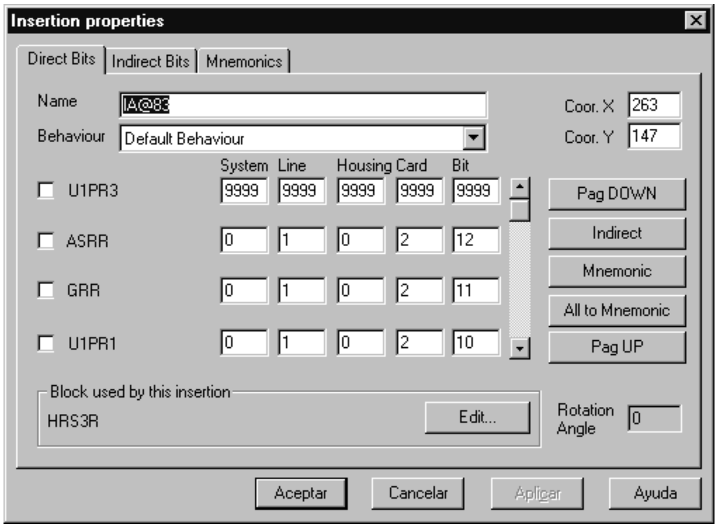
When conversion is finished, a log information is generated reporting about the conversion result. A typical log advising could be the next one:

```
@@ DtMC Global Conversion Log File - Date: 28/06/2001 - Time: 14:59
@@ GPC File: D:\DTMC\TESTS\Happy Rocks.GPC
@@ NO MGR file used!. Pure Direct to Mnemonic conversion

# START DtMC Conversion Log File.
# Number of Conversions: 135
# Number of Warnings: 13
# Number of Errors: 0
* Page: CH1A
  * Insertion: IA@1
    * Bit Comment: TRR
      Bit Comment converted FROM: LINE=1, HOUSING=0, CARD=2, BIT=7 TO NOT_ASSIGNED

  * Insertion: IA@2
    * Bit Comment: TRR
      Bit Comment converted FROM: LINE=1, HOUSING=0, CARD=2, BIT=7 TO NOT_ASSIGNED
* Insertion: IA@71
    * Bit Comment: ASRR
      Bit   Comment   converted   FROM:   LINE=1,   HOUSING=0,   CARD=0,   BIT=17   TO
NOT_ASSIGNED
    * Bit Comment: GRR
      Bit   Comment   converted   FROM:   LINE=1,   HOUSING=0,   CARD=0,   BIT=16   TO
NOT_ASSIGNED
    * Bit Comment: U1PR1
      Bit Comment converted FROM: LINE=1, HOUSING=0, CARD=0, BIT=3 TO NOT_ASSIGNED
    * Bit Comment: U1PR2
      WARNING:  Not  valid  direct  location  found:  LINE=9999,  HOUSING=9999,
CARD=9999, BIT=9999
      Bit Comment converted FROM: LINE=9999, HOUSING=9999, CARD=9999, BIT=9999 TO
NOT_ASSIGNED
    * Bit Comment: U1PR3
      WARNING:  Not  valid  direct  location  found:  LINE=9999,  HOUSING=9999,
CARD=9999, BIT=9999
      Bit Comment converted FROM: LINE=9999, HOUSING=9999, CARD=9999, BIT=9999 TO
NOT_ASSIGNED
```

Note that bit comments are passed to mnemonics and programmed as NOT_ASSIGNED. Note than warnings informing about non-direct bits found are stamped either.

Finally, button ***Save GPC*** saves the new changes in GPC file.

If user wants not only to pass from direct to mnemonic but also to assign mnemonics from a mnemonic list, button ***Directs > Assignation*** must be pressed instead of ***Directs > Mnemonics.*** A valid MGR file is required to do this conversion. An additional requirement is that GPC file has direct bits addressed with correct L,H,C,B format numbering, as for example the insertion shown in the figure 6.6:



Name: IA@83
Behaviour: Default Behaviour
Coor. X: 263
Coor. Y: 147

|  |  | System | Line | Housing | Card | Bit |
|---|---|---|---|---|---|---|
| □ | U1PR3 | 9999 | 9999 | 9999 | 9999 | 9999 |
| □ | ASRR | 0 | 1 | 0 | 2 | 12 |
| □ | GRR | 0 | 1 | 0 | 2 | 11 |
| □ | U1PR1 | 0 | 1 | 0 | 2 | 10 |

Block used by this insertion: HRS3R
Rotation Angle: 0

**Figure 6. 6 An example of L, H, C, B valid values.**

In the example figure, bit U1PR3 in unprogrammed and bits ASRR, GRR and U1PR1 have valid values.

As in the previous conversion, a log file is issued when detected non-direct bit comments in GPC file and invalid mnemonics in MGR file. The conversion is realised and log file and GPC file are generated.

Piece of information of a log file generated:

```
@@ DtMC Global Conversion Log File - Date: 28/06/2001 - Time: 15:19
@@ GPC File: D:\DTMC\TESTS\Happy Rocks.GPC
@@ MGR File: D:\DTMC\TESTS\road.mgr

## START DtMC MGR Log File due to WARNINGS.
## WARNING: The MGR file has 12 duplicated mnemonic/s:
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 12
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 27
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 28
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 29
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 30
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 55
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 56
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 57
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 58
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 59
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 60
DUPLICATED MNEMONIC FOUND --> DUMMY <-- AT POSITION: 74

## END DtMC MGR Log File.
## START DtMC Conversion Log File.
# Number of Conversions: 135
# Number of Warnings: 13
# Number of Errors: 0
* Page: CH1A
  * Insertion: IA@1
    * Bit Comment: TRR
      Bit Comment converted FROM: LINE=1, HOUSING=0, CARD=2, BIT=7 TO 9B.TRR
```

## 6.3 GPCCompare.

GPCCompare compares two GPC files. The comparation is done taking in to account GPC format information rather than line-to-line comparation as commercial comparation software did.

Next figures depicts the way of using GPCCompare Tool.

GPC Compare needs a template file to compare GPC files. To select a template you have to click on the “open template” button. Two template files are provided. One for GPC 2.0 format named **GPC_200_version2_1.tpl** and other for GPC 1.07 format named **GPC107_version1_0.tpl**.

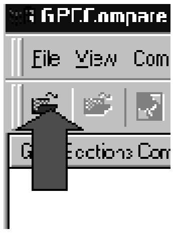


**Figure 6. 7 Loading template.**

When you have selected a template, you have to select the two GPC files that will be compared. Clicking on the “open GPC files” button does this operation.

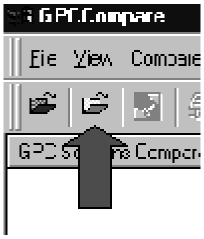


**Figure 6. 8 Loading GPC file.**

Now the application is ready for doing the comparison. Comparison starts when clicking on the “start comparison” button.

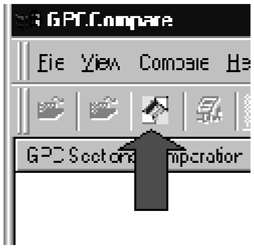


**Figure 6. 9  Starting comparation.**

Once the comparison is done. In the visualisation tree it can be seen all sections of the compared GPC files. Tree branches can be expanded by clicking on the “+” symbol in the visualisation tree.

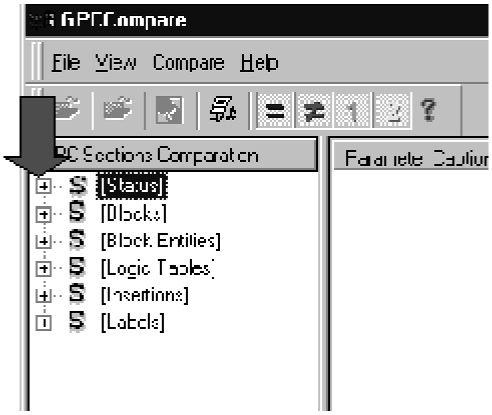


**Figure 6. 10  Tree view.**

When a section is expanded, all the items of that section will be shown in the visualisation tree.

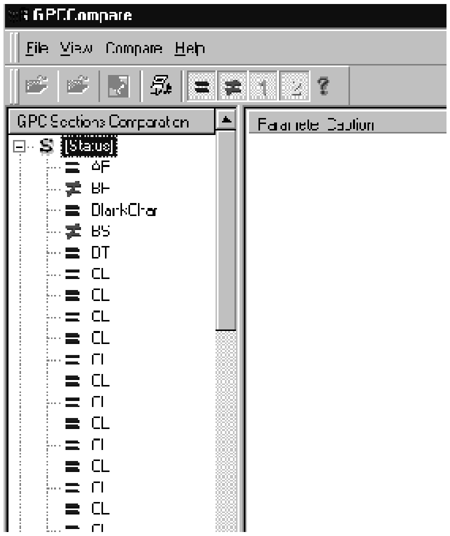
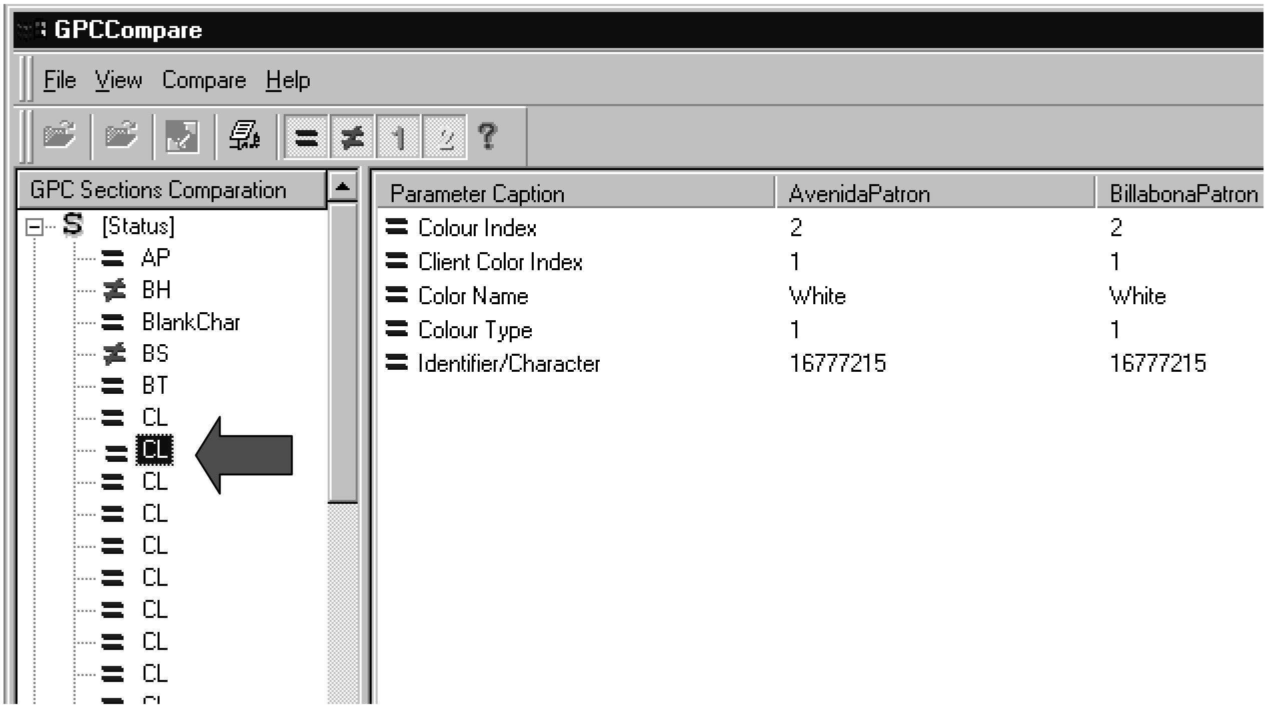


**Figure 6. 11  Expanded tree view.**

When a particular item is selected, the comparison of all the parameters of that item will be shown, parameter by parameter.



| Parameter Caption | AvenidaPatron | BillabonaPatron |
|---|---|---|
| = Colour Index | 2 | 2 |
| = Client Color Index | 1 | 1 |
| = Color Name | White | White |
| = Colour Type | 1 | 1 |
| = Identifier/Character | 16777215 | 16777215 |

**Figure 6. 12 Item comparation between two GPC files.**

Control on which items are displayed may be done using the buttons “display equal items”, “display different items”, “display items that only exist in GPC1”, “display items that only exist in GPC2” and “display not comparable items”. These buttons work as a filter, and they can be combined to obtain an optimal performance.

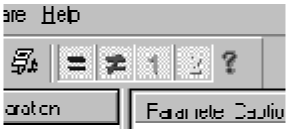
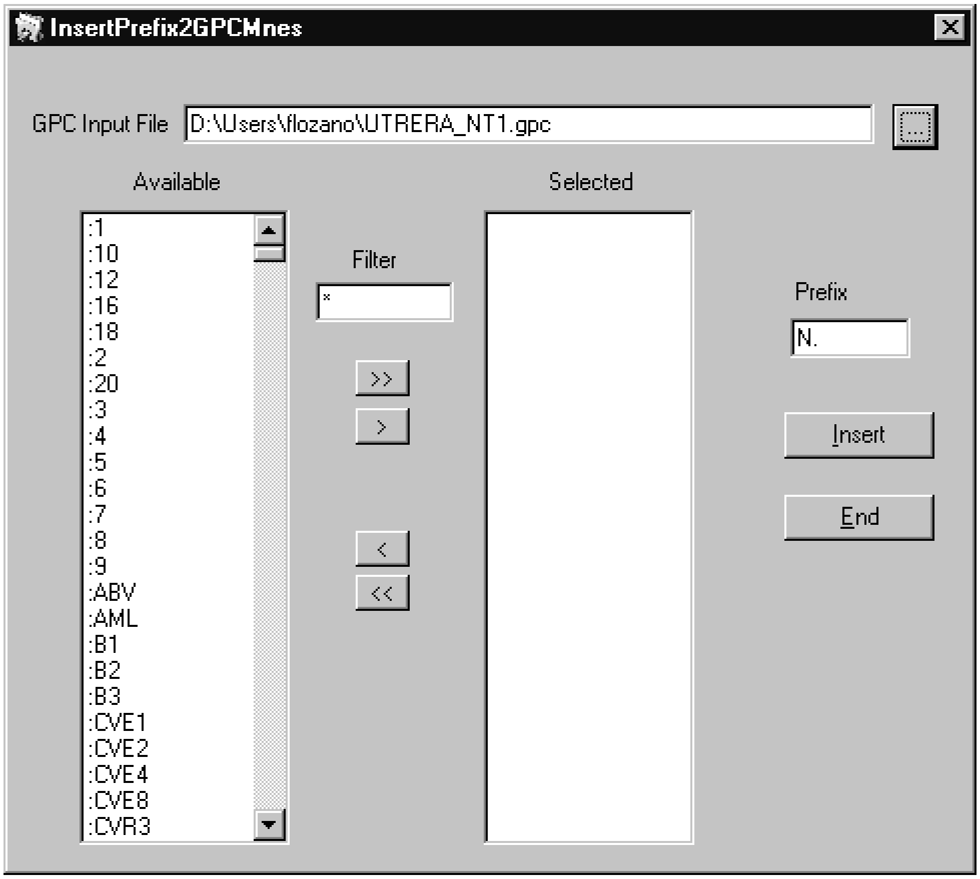


**Figure 6. 13 Comparation buttons.**

Finally, clicking the button “Report generation” can generate a report. Report will be generated according to the filter buttons active. (Usage of the filter buttons has been explained in the former picture).

## 6.4 InsertPrefix2GPCMnes

InsertPrefix2GPCMnes inserts prefixes to set of mnemonics of a selected file.



GPC Input File: D:\Users\flozano\UTRERA_NT1.gpc

Available:
:1
:10
:12
:16
:18
:2
:20
:3
:4
:5
:6
:7
:8
:9
:ABV
:AML
:B1
:B2
:B3
:CVE1
:CVE2
:CVE4
:CVE8
:CVR3

Filter: *

Prefix: N.

**Figure 6. 14 InsertPrefix2GPCMnes main window.**

The first step must be to select a GPC input file.

Once the input file has been selected, all its mnemonics are loaded into *available* list box, which it is in the left of the main window. At the moment, the user can be select a set of mnemonics to insert it a prefix.

There are several ways to select a set of mnemonics:

1. Use the following button to select all mnemonics:

   >>

2. The following button allows user to select a set of mnemonics. First, the mnemonics has been selected into available list box.

   >

3. If the set of mnemonics that will be selected follow the same pattern, the user can be use a filter to selected them. For that, write into *Filter* edit box the pattern and then push *Enter*.

A*

This operation moves mnemonics from *available* list box to *selected* list box.

The mnemonics within *selected* list box will be added the prefix into *prefix* edit box. The following figure shows the default value for prefixes.

N.

The mnemonics into *selected* list box can go back to *available* list box with the following buttons:

.- Go all the mnemonics back from *selected* list box to *available* list box.

.- Go the selected mnemonics back from *selected* list box to *available* list box.

The following button allows user to select a GPC output file and execute the tool.

To end the Tool, the user can use the following button:

## Apendix A — Known Limitations

### PCGE (Tested on Windows2000)

- Display of no True Type fonts may be erroneous.
- Mirror and scaling are not available in this version.
- MAA view printing option not available.
- Movement of many insertions/labels selected may produce jumps.
- Text's intensity of grey to indicate label selected is similar than other insertions. It provoke difficulties to visualize labels if selected and overlapped to other insertion.
- In block edition view. When a parallel entity is selected, properties view shows coordinates referred to the bottom left point of the parallel but the introduced value is the corresponding to the bottom left point of the bounding rectangle of the parallel.
- Some operations must be optimised in time.
- The PCGE Tool reads GPC files containing blocks and entities with the same name. In these cases the copy/paste operation is rejected. Normally, those files comes from older GPC versions.
- A program termination occurs when editing a block by double-click on block toolbar if it is in floating mode.
- When a label is moved no tracking is displayed.
- No insertions creation when block toolbar is in floating mode.
- Indexes of entities are not rearranged after using functions "Send to back" and "Bring to front. As a result, z-order does not save.
- GPC files with negative height empty rect entities are not loaded
- Some shortcuts exists with the same letter.
- Rotation toolbar is shown enabled when a label is selected, although no label rotation is permitted.
- "Do you want to save changes?" not shown after pasting between two gpc and after purging logic tables.
- ***Export…*** not available in this version.
- In some occasions, insertions indexes are not compacted when saving.
- Design view does not lose focus after changing from design view to MAA view.
- Some icons in toolbar must be deleted.

- Checks of bit comments in Insertion’s properties dialog are not “recalled” when page up and page down.

- Not show bit code for indirect label in Proeperties View.

- Not change common coordinates of a group of selected lements using Dynamic List.

- Paste operation permited after changing an extended logic table used in the clipboard.

- When pasting entities from a visible(invisible) button in other invisible(visible) block, a new behaviour from the origin block is added to target block.

- Copy/paste between heterogeneous views may provoke nonsense text pasted.

- Change blanckchar does not check if new blankchar is being used by mnemonics, so expressions asignment may be altered by this operation.

- Paralels with base width 0 are not allowed.

**GPCCompare**

- GPCCompare only compare two GPC files with the same version.

- When displaying the tree control (see user notes on the application's interface), a second level with item links is added. Only next links are available:

  - Blocks and their behaviour lists.

  - Block entity and its associated block.

  - Behaviours and their list of Logic Tables.

  - Logic Tables and their list of colours.

  - Insertions and their block.

  - Labels and their colours, and font.

- IMPORTANT: GPC Compare tool using tpl files "GPC_200_Version1_0.tpl" , "GPC_200_Version1_1.tpl" or "GPC_107_Version1_0.tpl" does not report differences in terms of z-order of Insertions or block Entities. This limitation is also applicable to GPC Compare Suite Tool 1.0 although it was not mentioned in the Release Record.

  Justification: Due to GPC format limitations, if some insertion/entity is sent to background or brought to front using PCGE functions all the relative position of Insertions/Entities will change. To avoid this side effect, the comparation of parameters insertion index between insertions, and entity index between blocks do not take in consideration the insertion index and the entity index respectively.

- IMPORTANT: GPC Compare tool using tpl files "GPC_200_Version1_1.tpl" or "GPC_107_Version1_0.tpl" reports that a label has been removed and that a label has been created if one label is moved. If using GPC_200_Version1_0.tpl the effect is that labels may be reported as created new if button "Send to back" and "Bring to front are used". In any case, the limitation is inevitable due to GPC format design.