

*GLG HMI Configurator  
User's Guide*

*GLG Toolkit  
Version 4.5*

**Generic Logic, Inc.**

---

**Generic Logic, Inc.**

6 University Drive 206-125

Amherst, MA 01002

USA

Telephone: (413) 253-7491

FAX: (413) 241-6107

email: [support@genlogic.com](mailto:support@genlogic.com)

web: [www.genlogic.com](http://www.genlogic.com)

**Copyrights and Trademarks**

Copyright © 1994-2025 by Generic Logic, Inc.

All Rights Reserved. This manual is subject to copyright protection.

GLG Toolkit, GLG Widgets, GLG Graphics Builder and GLG HMI Configurator are trademarks of Generic Logic, Inc.

All other trademarks are acknowledged as the property of their respective owners.

April 8, 2025

Software Release Version 4.5

# GLG HMI Configurator User's Guide

## Table of Contents

<b>Chapter 1</b>	<b>Using GLG HMI Configurator .....</b>	<b>11</b>
	HMI Configurator Layout.....	11
	Loading a GLG Drawing .....	12
	Creating a New Drawing.....	12
	Saving the Drawing.....	12
	Creating Objects.....	13
	Adding Predefined Components to the Drawing.....	13
	Creating Graphical Objects .....	13
	Magnetic Mode.....	14
	Editing GLG objects .....	14
	Selecting an Object and Changing Object Geometry.....	15
	Choosing from Several Selected Objects .....	16
	Using Undo.....	16
	Editing Object Properties .....	17
	Editing Public Properties of Prebuilt Widgets and Custom Components .....	19
	Editing Objects with the Edit Toolbox.....	20
	Multiple Selection .....	20
	Graphical Objects and Their Attributes .....	21
	Polygon Object.....	21
	Parallelogram.....	22
	Rounded Rectangle.....	22
	Arc Object .....	23
	Circle .....	24
	Ellipse.....	24
	Image Object .....	24
	Text Object.....	25
	Marker Object.....	27
	Spline Object .....	27
	Connector Object.....	27

GIS Object.....	28
Rendering Attributes.....	29
Drawing and Viewport Attributes.....	30
Integrated Zooming and Panning .....	31
Adding Geometrical Dynamics .....	32
Adding Geometrical Dynamics to Control Points .....	33
Adding Attribute Dynamics.....	33
Color List.....	34
Color Threshold.....	34
Color Blinking.....	34
Color Alert.....	35
Color Blinking Alert.....	35
List.....	36
Flow.....	36
Threshold.....	36
Range Alert .....	37
Blinking.....	37
Blinking Alert.....	37
Visibility Threshold.....	38
Value Display .....	38
Text Display .....	39
JavaScript .....	39
Paste Dynamics .....	39
Adding Tags for Data Supply .....	40
Adding a Tag to an Object Attribute .....	41
Mapping Tags to DataSources in the HMI Configurator .....	41
Browsing Tags Defined in the Drawing.....	42
Using Resources to Browse Objects.....	42
Prototyping Dynamic Behavior Using Run Mode .....	44
Defining Alarms.....	44
Range Alarm .....	45
Range2 Alarm .....	45
Change Alarm .....	45
Browsing Alarms.....	46
Using GLG Widgets and Palettes.....	46
Object Layout and Alignment .....	47
Defining Object Tooltips.....	49
<b>Chapter 2   GLG HMI Configurator Menus .....</b>	<b>51</b>
File.....	51
New Drawing (Resize and Stretch).....	51
New Drawing (Resize, No Stretch).....	52

---

New Drawing (Fixed Scale, No Stretch).....	52
Reset Drawing .....	53
Work Area Size .....	53
Unset .....	53
Custom Size .....	53
Size From Config File.....	53
Adjust \$Widget Size.....	53
To Visible Area.....	53
To Work Area Size .....	53
Set Custom Size (Pixels).....	53
Resize Drawing .....	53
Open .....	54
Open URL .....	54
Load Template.....	54
Recent Drawings .....	54
Save .....	54
SaveAs.....	55
Print (Windows and GTK on Linux only).....	55
Export PostScript.....	55
Print Settings .....	55
Page Layout.....	55
Stretch .....	56
Center.....	56
PostScript Level .....	56
PostScript Orientation.....	56
Save Image .....	56
Save Image Full.....	56
Export Strings.....	57
Import Strings.....	57
Export Tags .....	57
Import Tags .....	57
Exit .....	57
Palettes .....	57
Manual Widget Positioning.....	58
Adjust Widget Size.....	58
Make Widget Viewports Transparent .....	58
Use Light Viewports.....	58
Custom Objects .....	59

Custom Widget Samples .....	59
Read Palette .....	59
Read Directory .....	59
Edit .....	59
Select Multiple Objects .....	59
Select Rectangular Area .....	60
Select All .....	60
Cut .....	60
Copy .....	60
Paste .....	60
Delete .....	60
Define Clone Offset .....	60
Copy Object With an Offset .....	61
Edit Group Elements .....	61
Set D Resource .....	61
Set G Resource .....	61
Set S Resource .....	61
Recursive .....	61
Reset Scaling Xform .....	62
Run-Time Scrollbars .....	62
None .....	62
X and Y .....	62
X .....	62
Y .....	62
Keep Edit Ratio .....	62
Undo .....	62
Undo .....	62
Undo History .....	62
View .....	63
Zooming .....	63
Zoom In .....	63
Zoom Out .....	63
Zoom To .....	63
Set Zoom Factor .....	63
Reset Zoom .....	63
Pan To .....	63
Scroll by Dragging .....	64
Traverse .....	64
Set Focus .....	64

---

Main Focus .....	64
Arrange.....	64
Create Temporary Group.....	64
Select Multiple Objects .....	64
Add Object to Group .....	65
Delete Object from Group .....	65
Add or Delete Object from Group.....	65
Select Next.....	66
Edit All (First) .....	66
Edit All (Select).....	66
Permanent Group.....	67
Explode Group.....	67
Reorder .....	67
Move to Back .....	67
Bring to Front.....	67
Move Backward .....	67
Move Forward.....	67
Convert Viewport .....	68
Viewport -> Light Viewport .....	68
Light Viewport -> Viewport .....	68
UTF8 Conversions.....	68
Convert Strings from Current Local to UTF8.....	68
Convert UTF8 Strings to Current Locale.....	68
Layout .....	68
Layout Toolbox .....	68
Position in Drawing.....	68
At Left Edge.....	68
Center Horizontally.....	68
At Right Edge.....	69
At Top Edge .....	69
Center Vertically .....	69
At Bottom Edge .....	69
Set Width/Height.....	69
World Coordinates .....	69
Pixels.....	69
Set Space/Distance .....	70
World Coordinates .....	70
Pixels.....	70
Align .....	70
Align Left.....	70
Align Horiz. Center.....	70

Align Right.....	70
Align Top.....	71
Align Vert. Center.....	71
Align Bottom.....	71
Make Same Size.....	71
Width.....	71
Height.....	71
Both.....	71
Distribute.....	71
Across.....	71
Down.....	71
Space Evenly.....	71
Across.....	72
Down.....	72
Distribute Evenly.....	72
Across.....	72
Down.....	72
Select Anchor.....	72
Align Points.....	72
Align To Viewport Span.....	72
More.....	73
Object.....	73
Create.....	73
Polygon.....	73
Rectangle.....	73
Parallelogram.....	74
Rounded Rectangle.....	74
Arc.....	74
Circle.....	75
Ellipse.....	75
Spline.....	76
Marker.....	76
Image.....	76
Text.....	76
GIS Object.....	77
Connector.....	77
Edit Toolbox.....	78
Properties.....	78
Public Properties.....	78
Resources.....	78
Tags.....	78
Alarms.....	78
Object Dynamics.....	79

---

Add Dynamics.....	79
Edit Dynamics.....	79
Tooltip .....	79
<i>Add Tooltip</i> .....	79
Edit Tooltip .....	79
Delete Tooltip .....	79
Actions.....	79
Add Command .....	79
Add Input Command.....	80
Edit Actions.....	80
Delete All Actions.....	80
Mark Actions.....	80
Paste Actions.....	80
Disable User Interaction .....	80
Run.....	80
Start.....	80
Stop.....	81
Options.....	81
Draw Grid.....	81
Snap To.....	81
Show Axis .....	81
Show Coordinates.....	82
Show Span .....	82
Span Marks .....	82
Span Area.....	82
Save Format.....	82
Save Compressed.....	82
SaveImageFormat.....	82
Convert Strings to UTF8 on Save .....	83
Selection Options.....	83
Show Attachment Points.....	83
Disable Dynamics For Editing.....	83
Select Non-Components .....	83
Trace / Highlight.....	83
Trace Tags.....	83
Trace Resources & Constraints.....	83
Highlight Widgets .....	84
Highlight Top Level Objects.....	84
Magnetic Control Points.....	84
Attach to Point .....	84
Use Point Position.....	84
Disable .....	84

Color Options.....	84
Swap Color Palettes .....	84
255 Color Display .....	84
Dynamics Options.....	84
Treat Dynamics as Global on Copy .....	84
Advanced Dynamics Display .....	85
Data Browser Options.....	85
Browse Tag Names .....	85
Browse Tag Sources.....	85
Appearance.....	85
Scrollbars.....	85
Toolbar .....	85
Force File Extensions.....	85
Display OpenGL Info.....	85
Help.....	85
Online Reference.....	85
About GLG Toolkit.....	86

## Chapter 1

# Using GLG HMI Configurator

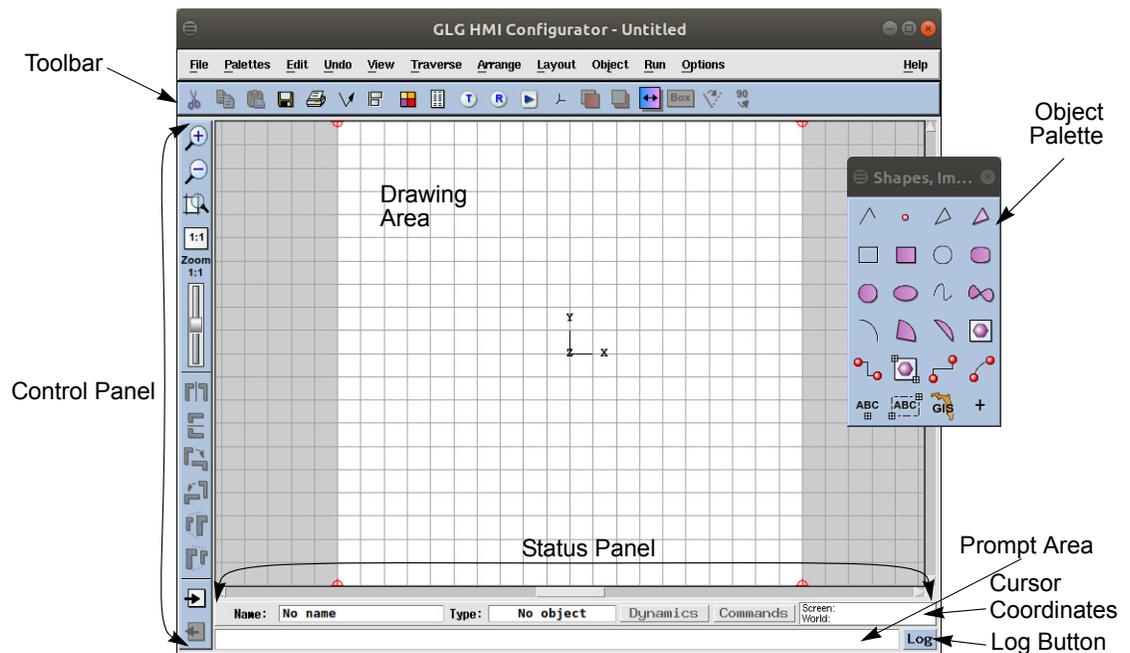
# 1

The GLG HMI Configurator is a simplified drawing editor intended for end users to edit the application's drawings or create new drawings. The HMI Configurator enables you to add new objects to a drawing, as well as edit attributes of the existing objects in the drawing. The HMI Configurator's Run Mode can be used to test run-time animation of the drawing.

The first part of this document presents a tutorial-style overview of the tasks involved in editing drawings. The second part provides a formal list and description of all options on the GLG HMI Configurator's menus.

## HMI Configurator Layout

When the HMI Configurator starts, you see the main window:



The HMI Configurator window contains the following areas:

The **Toolbar** contains icon buttons for fast access.

The **Drawing Area** contains the graphical objects of the drawing. By default, the drawing area contains an axis marker that shows the center of the coordinate system, and grid lines.

The **Object Palette** contains buttons for creating graphical objects.

The **Control Panel** contains zooming, panning and other controls.

The **Status Panel** contains two rows of controls:

- The top row displays the name and type of the selected object; it also contains buttons for

quick access to the geometrical dynamics and command actions attached to the object, if any. On the right of the top row there is a control that displays the cursor position in both screen and world coordinates.

- The bottom row of the *Status Area* contains the *Prompt Area* that displays messages and prompts for action; it also contains the *Log* button that displays a log of messages generated by the current editing session.

## Loading a GLG Drawing

A GLG drawing may be loaded into the HMI Configurator using the *File, Open* menu option. On Windows, the drawing may also be loaded by double-clicking on the drawing file.

A GLG drawing is a collection of objects contained in a viewport named “*\$Widget*”. If the “*\$Widget*” viewport is missing, the drawing can not be loaded into the HMI Configurator. When the drawing is loaded into the HMI Configurator, the “*\$Widget*” viewport is hidden from the user and its content is displayed in the drawing area.

## Creating a New Drawing

When the HMI Configurator is first started, it automatically creates a new empty drawing.

The Configurator’s *New* menu also provides several options for creating new drawings:

- *New Drawing (Resize and Stretch)* creates a resizable drawing that stretches or changes the size of objects in the drawing when the drawing’s window is resized. Several suboptions allow to create drawings with different aspect ratio matching the aspect ratio of the display: *1:1*, *4:3* and *16:9*.
- *New Drawing (Resize, No Stretch)* creates a drawing that changes the size of objects in the drawing when the drawing is resized but does not stretch the objects, maintaining the X/Y aspect ratio. The suboptions allow to create drawings with aspect ratio matching the aspect ratio of the display: *1:1*, *4:3* and *16:9*.
- *New Drawing (Fixed Scale, No Stretch)* creates a drawing that does not stretch or resize the objects in the drawing, but shows a smaller or bigger part of the drawing area when the window is resized. The two suboptions provide choices for specifying the size of the drawing: *Custom Size* allows to specify the width and height of the drawing in pixels, and the *Size From Config File* option uses the size specified in the *glg\_hmi\_config* file.

An empty drawing created by the HMI Configurator on start up uses the *New Drawing (Resize and Stretch), 1:1* option.

## Saving the Drawing

To save the drawing, use either the *Save*  toolbar button or *File, Save* from the main menu.

The *Options* menu contains settings that control how the drawing is saved. The default *ASCII* save format enables the drawing to be displayed on different hardware platforms, as well as in the Java, C# and JavaScript versions of the Toolkit. The default *Save Compressed* option minimizes the size of the drawing.

The drawing is contained in a viewport named “\$Widget”, which is hidden from the user but is included in the drawing file when the drawing is saved.

## Creating Objects

There are several types of available objects in the HMI Configurator:

- **Prebuilt GLG Widgets** available in the Palettes menu. These widgets provide a list of **public properties**, which make it easier to modify the widget’s appearance and behavior
- **Predefined Components** provided by a OEM vendor are prebuilt components with the application-specific appearance and dynamic behavior. A predefined component may provide a list of **public properties** for simplified editing.
- **Simple graphical objects** may be created using the *Shapes, Images and Text* palette and include polygons, arcs, text objects, images and other graphical primitives. The appearance of graphical primitives is defined by their **attributes** (or **properties**).

### Adding Predefined Components to the Drawing

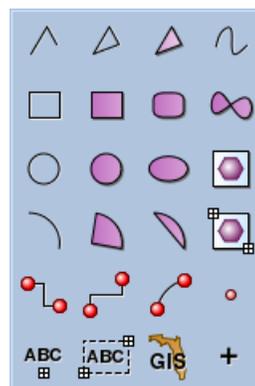
Prebuilt GLG Widgets and predefined components supplied by an OEM vendor are added to the *Palettes* menu. The *Palettes, Custom Widget Samples* menu option provides a palette containing samples of predefined components with defined public properties. To add a widget or component to the drawing, simply click on its icon in the palette, then position it with the mouse.

Use *Object, Properties* to inspect the component’s public properties.

By default, *Object, Properties* displays public properties of an object or component in the *Public Properties* dialog. If an object or component does not have public properties defined, its attributes will be displayed in the *Properties* dialog. Refer to the *Editing Public Properties of Prebuilt Widgets and Custom Components* chapter on page 19 for more information.

### Creating Graphical Objects

You can create graphical objects by selecting on of the drawing primitives in the *Object Palette* on the left of the *Drawing Area*:



Object Palette

All buttons in the HMI Configurator have tooltips, so you can move the mouse over a button to find its exact function.

To create an object, select one of the drawing primitives on the left (a *Filled Polygon* , for example) and click several times in the drawing to define the polygon's points. To finish, press the right mouse button on Windows, the middle mouse button on Linux/Unix, or the *Escape* key on both platforms.

For different objects, a different number of control points may be required. For example, an arc will require 3 control points.

Some objects may also require entering additional information. For example, you'll be asked to type a text string when creating a text object.

When objects are being created or any other actions requiring user input are performed, a prompt with additional instructions is displayed in the *Prompt Area* at the bottom.

### ***Magnetic Mode***

Some symbols, such as widgets from the process control or electrical widget sets, have attachment points that can be used for attaching lines representing pipes or wires. The Magnetic Mode is enabled by default to automatically attach pipes and lines to selected points.

When creating a polygon, moving a mouse over an object (such as a tank) makes all object's attachment and control points visible and highlights a point next to the mouse position. Clicking on the highlighted point will attach the end of the polygon to that point. The user can use *Control-click* to position the polygon point in the vicinity of the highlighted point without constraining.

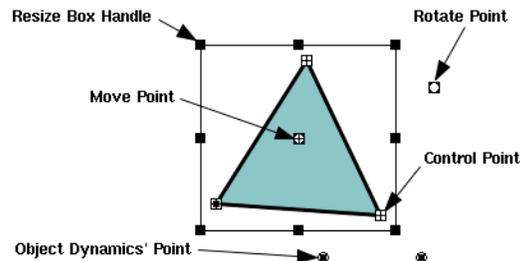
The *Options, Magnetic Connection Points* menu may be used to either disable Magnetic Mode, or modify it to use the point's position without attaching to it. The Magnetic Mode default can also be changed using the *MagneticMode* variable in the *glg\_hmi\_config* file.

### ***Editing GLG objects***

After an object is created, it may be edited in a variety of ways. To edit an object, it has to be selected.

## Selecting an Object and Changing Object Geometry

The simplest way to select an object is to click on it with the left mouse button. When an object is selected, its control points and resize box are highlighted in the drawing, and its name and type are displayed in the *Status Panel* at the bottom.



Object Selection: Control Points

**Control points** appear at the vertices or other important points. To change the shape of the object, drag its control points with the mouse. The *Control Point* dialog for precise editing of a control point can be activated by *Shift*+clicking on the control point.

Control Point Dialog

The control point's position can be adjusted using the directional arrows, or it can be defined by entering the point's coordinates into the *Value* field of the dialog. To position the control point at the same location as another control point, click on the *Use Point* button, then click on another point to reuse its position.

If more than one control point is potentially selected by the *Shift*-click, the selection can be rotated using the arrow buttons in the dialog's upper right corner. The dialog also contains buttons for adding control point dynamics, tag and value monitoring alarms for advanced use. The dialog's *Name* field is a read-only and can not be changed in the HMI Configurator.

You can reposition an object by simply dragging it with the mouse. The **move point** at the object's center may be used to position the object precisely by *Shift*+clicking on the move point and using the arrows in the *Move Handle* dialog. The dialog's *Position* text field may be used to position the object by specifying the new coordinates of its center.

An 8-point **resize box** appears around an object. Use its points to resize the object. Objects with only one control point (markers, fixed text and fixed size images) can't be resized, and resize points for these objects appear desensitized (in a gray color). *Shift*+click on a resize point to position it precisely with the arrows in the *Resize Handle* dialog.

A **rotate point** appears on the right side on the resize box. To rotate an object, drag the rotate point with the mouse. To rotate an object precisely by a specified angle, *Shift*+click on the rotate point and use the arrows in the *Rotate Handle* dialog. For quicker or less precise rotation, drag the rotate point with the mouse.

If the object has geometrical dynamics attached, the **Dynamics' Points** (such as Start Point, End Point or Rotation Center) may also be displayed depending on the setting of the *ControlPointDisplay* variable in the *glg\_hmi\_config* configuration file.

To select an object with no fill (for example, an unfilled polygon), click on the object's edge. The *FillType* attribute controls this aspect of an object's appearance. To avoid accidental movement while you are selecting an object, use *Shift*+click to select the object.

Press the *Escape* key to unselect.

### ***Choosing from Several Selected Objects***

When several objects are located close to each other and it is difficult to select the object of your choice, you can use the *Shift* key to help choose among several objects.

*Shift*-clicking in the drawing area with the left mouse button pops up a menu that allows you to select an object out of several potentially selected objects.

If the *Properties* dialog is open, the arrow buttons in the upper right corner of the dialog may be used to select an object when several objects are potentially selected.

### ***Using Undo***

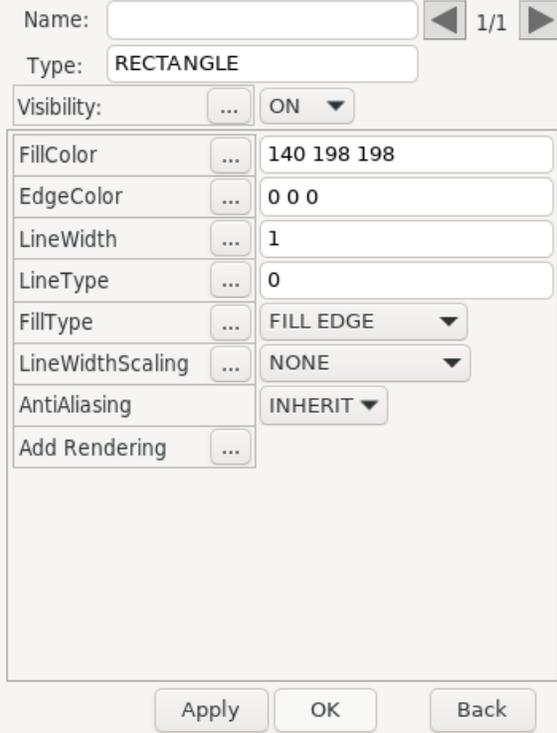
The changes applied to the object's geometry or attributes may be reverted by selecting *Edit, Undo* options from the main menu, where the last operation is displayed at the top of the menu. For example, try moving an object with the mouse. To undo the move operation, select *Edit, Undo Object Move or Stretch*.

For multiple step undo, *Edit, Undo History* option may be used. For example, try resizing the object using its resize box and then click and drag one of its control points with the mouse. To revert one or both of these operations, select *Edit, Undo History, Undo Point Move* to undo the point move operation, and then select *Edit, Undo History, Undo Object Move or Stretch* to undo the resize operation. Notice that the *Undo History* list displays performed operations in the reverse order, so that the last operation appears at the top of the list.

The order of undo steps is important and should be considered when reverting a complex sequence of editing operations.

### Editing Object Properties

You can use the *Properties*  toolbar button to bring the *Object Properties* dialog. Alternatively, you can use the right mouse button and select *Properties* from a popup menu, or *Object, Properties* from the main menu.



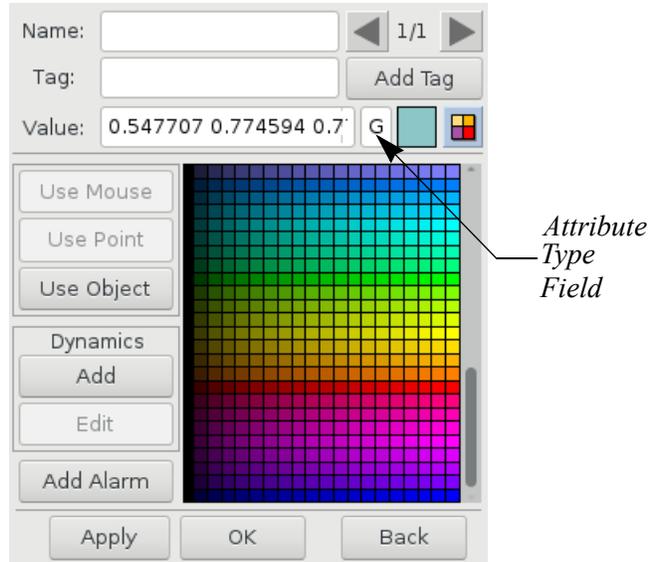
Name:	<input type="text"/>	◀ 1/1 ▶
Type:	RECTANGLE	
Visibility:	<input type="checkbox"/>	ON ▼
FillColor	<input type="text"/>	140 198 198
EdgeColor	<input type="text"/>	0 0 0
LineWidth	<input type="text"/>	1
LineType	<input type="text"/>	0
FillType	<input type="text"/>	FILL EDGE ▼
LineWidthScaling	<input type="text"/>	NONE ▼
AntiAliasing	<input type="text"/>	INHERIT ▼
Add Rendering	<input type="text"/>	
Apply   OK   Back		

Object Properties Dialog

The *Object Properties* dialog allows you to edit an object's properties, or attributes. The terms properties and attributes will be used interchangeably in this document. We will refer to the *Object Properties* dialog simply as the *Properties* dialog in the rest of this manual.

Some generic properties common for all objects, such as *Name* and *Visibility*, are displayed at the top of the dialog. Specific properties that depend on the type of the selected object are listed below.

Some properties, for example the *FillColor* property, have an ellipsis button  next to them. The button indicates that the attribute is an object, and you can press it to bring the *Properties* dialog with a color palette and other options for editing the attribute. We will refer to this dialog as the *Attribute Properties* dialog in the rest of this manual to differentiate it from the *Object Properties* dialog.



Attribute Properties Dialog

You can select a new fill color from a palette or specify an RGB value in the *Value* field. You can also select colors from a custom OEM color palette: *Ctrl-click* on the color palette to swap the palettes.

You can also *Shift-click* in the color palette or select *Options*, *Color Options*, *Pastel Colors* to switch between pastel and regular colors.

By default, the color RGB values are displayed in the [0;1] range, but it may be changed to the [0;255] range by selecting *Options*, *Color Options*, *2555 Color Display* option in the main menu. To permanently use the 255 color range, set the *ColorDisplay255* variable in the *glg\_hmi\_config* configuration file.

For other attributes, different attribute-related palettes will be shown: try selecting the ellipsis button  for the *LineWidth* and *LineType* attributes.

For the *TextString* attribute of a text object, the dialog will contain a text edit box for entering a multi-line text, while the text box in the *Object Properties* dialog allows only a single-line text string to be entered.

The *Type* field of the *Attribute Properties* dialog shows the type of the attribute object and may be *D* for double numerical values, *S* for string values or *G* (geometrical) for XYZ or RGB triples used to represent colors and control points.

The *Name* field is read-only and cannot be changed in the HMI Configurator. The read-only *Tag* field displays tag information. Refer to the *Adding Tags for Data Supply* section for information on adding and editing tags.

To reuse the attribute's value from another object in the drawing, click on the dialog's *Use Object* button, then click on the object whose attribute value you want to use. For example, if the dialog displays the *FillColor* property, using the *Use Object* button will reuse another object's *FillColor*. The *Attribute Properties* dialog also has buttons for adding attribute dynamics, tag and value monitoring alarm described in the following sections.

### ***Editing Public Properties of Prebuilt Widgets and Custom Components***

Prebuilt GLG Widgets available in the Palettes menu provide a list of public properties, which make it easier to modify the widget's appearance and behavior. Predefined components supplied by an OEM vendor may also have public properties for convenient editing.

By default, the HMI Configurator automatically selects what properties will be displayed in the object's *Properties* dialog. If the selected object has public properties, the *Properties* dialog will present a simplified view of the object by listing only its public properties and hiding all other attributes of the object.

If the selected object does not have public properties, the *Properties* dialog will list all attributes of the object.

The HMI Configurator can be configured (via the *glg\_hmi\_config* configuration file) to enable two property panels: one for the object attributes and another for public properties. In this case two *Property* icons will be present in the toolbar, and the *Object* menu will have *Properties* and *Public Properties* entries. The HMI Configurator can also be configured to disable editing of the predefined components other than changing their public properties.

The *Custom Widget Samples* palette provides samples of predefined components. Select *Palettes*, *Custom Widget Samples* and select the tank component. Select *Object*, *Properties* from the main menu or use the *Properties* toolbar icon to display the *Properties* dialog containing the list of the component's public properties. For easier editing, some properties in the *Public Properties* dialog are sorted into categories, such as *Body* or *Label*. Each category has a corresponding dialog button to access properties in that category.

The ellipses button  next to each property brings a dialog for editing the property. This dialog is similar to the dialog for editing object attributes described above.

## Editing Objects with the Edit Toolbox

The *Edit Toolbox* can be used as an alternative to the *Properties* dialog for quick and easy editing of an object's visual appearance, as well as editing multiple objects or all objects in a group. Select a polygon in the drawing and click on the *Edit Toolbox*  toolbar button to bring the *Edit Toolbox*.



Edit Toolbox

You can edit various attributes, such as *FillColor* or *LineWidth*. For example, click on the *LineWidth* icon and select a new line width from the line width palette. Notice that the text box at the bottom of the dialog reflects the current setting for the selected attribute.

Only the attribute icons that are applicable to the selected object type are activated in the *Edit Toolbox*, while the other icons are disabled. For example, for the polygon object, the *FillColor* and *EdgeColor* icons are activated, while the *TextColor* buttons are disabled. The *Edit Toolbox* does not display all attributes of an object but only the most commonly used attributes of a given object type.

## Multiple Selection

Create three polygons using the *Filled Polygon*  button. To select multiple objects for editing, click and drag a rectangle to enclose the objects you want to edit. For example, click and drag a rectangle to include all three polygons. This creates a temporary group that includes all selected objects.

Notice that the *Status* panel at the bottom of the HMI Configurator's window indicates that the selected object is of type *Group* and its name is *\$TempGroup*, which is a predefined name for a temporary group. The resize box of a temporary group is shown as a dashed line. A temporary group is volatile and will be discarded when it is unselected.

To include or exclude an object from the group, *Ctrl+click* on the object you want to delete or add. For example, *Ctrl+click* on one of the polygons to delete it from the group. *Ctrl+click* on the same polygon again to add it back to the group.

To edit properties of three polygons, click on the *Edit Toolbox*  toolbar button to bring the *Edit Toolbox*. Click on the *FillColor* button in the *Edit Toolbox* and select a new color in the color palette. All selected objects will reflect the color change.

In the above example, a temporary group was created using a click and drag operation that defined a rectangular area containing the objects to be selected. This option may be inconvenient in some cases, when there are several intersecting objects and precise object selection is required. Alternatively, a temporary group may be created by *Ctrl+clicking* on each object that needs to be selected.

For example, *Ctrl+click* on one of the polygons with the left mouse button to select it. *Ctrl+click* on another polygon, and then on the third one. All three polygons will be selected. Attribute editing of all selected objects can be done using the *Edit Toolbox*, as described above.

Press the *Escape* key to unselect the objects.

## Graphical Objects and Their Attributes

While the attributes of graphical objects differ depending on the object type, the *Name* and *Visibility* attributes are common for all graphical objects. The *Name* attribute keeps a user-assigned name which can be used to identify the object at run time.

The *Visibility* attribute can be used to make the object invisible or to implement blinking dynamics. A value of 1 indicates that the object is visible; a value of 0 makes the object invisible. In the OpenGL version of the Configurator, as well as in the GTK version on Linux, the values between 0 and 1 make the object transparent, with the object transparency increasing as the visibility value decreases. The negative values between 0 and -1 may be used to dim the object.

Optional *Rendering Attributes*, such as gradient fill, cast shadows, arrowheads and fill dynamics, can be added to most of the graphical objects as described in the *Rendering Attributes* section on page 29.

The rest of this chapter describes attributes of graphical primitives available in the HMI Configurator.

Refer to the *GLG Objects* chapter of the *GLG User's Guide and Builder Reference Manual* for description of attributes of other object types as well as detailed description of all object attributes.

### Polygon Object

In GLG, the polygon is a fundamental graphical primitive. Most other GLG graphical objects (arcs, rectangles, etc.) inherit their attributes from the polygon object.

To create a filled polygon, click on the filled polygon icon , then click in the drawing area to define control points. Press the *Escape* key to finish. The open polygon  and closed polygon  icons create open or closed polygons with lines and no fill.

To edit the polygon's properties, use *Object, Properties* from the main menu or click on the *Properties*  toolbar icon. It will display the polygon's *Properties* dialog containing the following properties:

- The *FillColor* and *EdgeColor* attributes control the polygon's colors. To select the fill color

from a color palette click on the ellipsis button  for the *FillColor* or *EdgeColor* attribute and select a color from a color palette. The colors may also be entered as RGB values by typing them into the corresponding *Value* fields of the *Properties* dialog. You can also select colors from a custom OEM color palette: *Ctrl-click* on the color palette to swap the palettes.

By default, the color RGB values are displayed in the [0;1] range, but it may be changed to the [0;255] range by setting the *ColorDisplay255* variable in the *glg\_hmi\_config* configuration file.

- The *LineWidth* defines the line width in pixels. To select a line width from a palette, click on the ellipsis button  for the *LineWidth* attribute.
- The *LineStyle* attribute defines the line type. Use the ellipsis button  for the *LineStyle* attribute to display a palette of available line types.
- The *FillType* attribute controls whether the polygon is drawn as an outline, fill or both. In addition to combination of the *EDGE* and *FILL* values, it also has a line fill (*LFILL*) value that is used to draw thick lines with two colors, using *EdgeColor* for the line's edge and *FillColor* for the middle part of the line.
- The *OpenType* attribute controls whether the polygon draws the line connecting the first and last polygon's points.

The last item in the property list allows adding and editing optional rendering attributes, such as gradient fill, cast shadows, arrowheads and fill dynamics. These attributes are described in the *Rendering Attributes* section on page 29.

## Parallelogram

The parallelogram object is used to create rectangles. To create a parallelogram, click on the rectangle  or filled rectangle  icon and select two points. It creates a parallelogram with three control points which define its shape.

The attributes of a parallelogram are the same as the attributes of a polygon except that the *OpenType* attribute is not present.

## Rounded Rectangle

This object is used to create rectangles with rounded corners. To create a rounded rectangle, click on the filled rounded rectangle  icon and select two points. It creates a parallelogram with rounded corners and three control points which define the object's shape.

The rounded rectangle inherits all polygon attributes like *LineStyle* and *FillColor*. In addition, a rounded rectangle has the following attributes:

- *Radius1*  
Controls the vertical extent of the rounded corners.
- *Radius2*  
Controls the horizontal extent of the rounded corners. If set to -1, the value of *Radius1* is used, and the size of rounded corners may be controlled with a single parameter - *Radius1*.
- *UnitType*

- Specifies units used for *Radius1* and *Radius2* attributes and may have the following values:
- SCREEN COORDINATES - corner radiuses are defined in screen coordinates; the size of the rounded corners stays constant.
  - WORLD COORDINATES - corner radiuses are defined in world coordinates; the size of the rounded corners changes proportionally when the drawing is resized, but stays constant when the object is resized with the mouse.
  - RELATIVE - corner radiuses are defined as coefficients in the [0;1] range relative to the extent of the rectangle's corresponding side. The size of the rounded corners changes proportionally when the drawing or the object is resized, maintaining a constant ratio between the size of the rounded corners and the length of the object's side in the corresponding direction.

- *Resolution*

The number of line segments used to render rounded corners. The default value is 7. A higher value may be used for nicer rendering of rectangles with large corner radiuses.

## Arc Object

The Arc object is used to represent both **arcs** and **circles**. There are two types of arcs:

- a **chord** arc  simply joins the two ends of the curve with a straight line
- a **sector** arc  is shaped like a piece of a pie, with two straight lines joined at the center of the arc.

A circle  is simply a special case of an arc whose angle is 360°.

To create an arc, click on one of the arc icons in the *Object Palette*, click in the drawing area to define the center of the arc, move the mouse and click to define its radius and the start angle, then move the mouse and click again to define the end angle. When creating a circle, only the center and the radius need to be defined.

Arcs and circles have two control points: a center and a vector point. These points coincide when an arc is created. You never need to edit these points, unless you want to change the arc's orientation in space. Make sure not to use the arc's control points to move an arc, otherwise it will cause the arc to rotate in 3D space. To move an arc, simply drag it with the mouse.

The arc inherits all polygon attributes like *LineType* and *FillColor*. In addition, an arc has the following attributes:

- *ArcFillType* defines the type of the arc: *chord*  or *sector* .
- *StartAngle* and *EndAngle* define the start and end angles of the arc. The angles are measured in degrees (counter clockwise) from the 3 o'clock position.
- *Radius* defines the arc's radius.
- *Resolution* specifies the number of line segments used to render the perimeter of an arc or a circle. A circle drawn with a resolution of 5 is simply a regular pentagon. The default resolution is 100. A smaller value may be used for small arcs and circles to increase performance when the drawing contains a large number of arcs.

## Circle

A circle  is simply a special case of an arc whose angle is 360° and has the same attributes.

To create a circle, click on one of the circle icons in the *Object Palette*, click in the drawing area to define the center of the circle, then move the mouse and click to define its radius.

Circles have two control points: a center and a vector point. These points coincide when a circle is created. You never need to edit these points, unless you want to change the circle's orientation in space. Make sure not to use the circle's control points to move a circle, otherwise it will cause the circle to rotate in 3D space. To move a circle, simply drag it with the mouse.

## Ellipse

An ellipse  is a special case of a rounded rectangle with relative radiuses set to 1 and has the same attributes.

To create an ellipse, click on the filled ellipse  icon and select two points. It creates a parallelogram with three control points which define the ellipse's bounds.

## Image Object

The image object is used to embed images in the GIF, JPEG, PNG and BMP format. The GIF, JPEG and PNG formats are supported across all platforms, while the BMP format is supported only in the C/C++ version on Windows. Transparent colors are supported via the *TransparentColor* attribute, and image transparency is supported via the *Visibility* attribute (by setting it to fractional values).

An image may be of **fixed** size, defined by the size of the image in the original file, and have one control point. The image object may also be **scalable**, in which case it's size is adjusted to fit the rectangle defined by the image's two control points.

To create a fixed size image, click on the fixed image  icon, then click in the drawing area to define the image's control point. To create a resizable image, use the scalable image icon  and define two control points that define the image's area. An image object has the following attributes:

- *ImageType* defines the type of the image: a fixed-size image with 1 control point or a scalable image with two control points.
- *ImageFile* defines the location of the image file in the GIF, JPEG, PNG or BMP format. The type of the file is determined by the file's extension: *.gif*, *.jpg*, *.png* or *.bmp*. Using relative image paths is recommended for more flexible deployment.
- *Anchor* defines the vertical and horizontal alignments of the fixed size image relative to its control point. The choices are CENTER, LEFT, or RIGHT for horizontal alignment, and CENTER, TOP, or BOTTOM for the vertical. The attribute has no effect for scalable images.
- *TransparentColor* defines the image color to be rendered as transparent. This may be used for rendering icons with transparent background. The color RGB values are specified using the default 0-1 range, or using 0-255 range if the *255 Color Display* option is activated. All image pixels with this RGB color value will be rendered as transparent.

By default, the attribute is set to a value which disables transparency: (-1,-1,-1) in the default color mode, or (-255,-255,-255) value in the *255 Color Display* mode. For transparent GIF images, the *TransparentColor* setting overrides the transparent color defined in the GIF file. If *TransparentColor* is set to the disabled value described above, the transparent color defined in the GIF file is used.

**Windows Note:** the transparent color mode is supported on OS versions that support TransparentBlit method.

## Text Object

The text object is used to place labels and legends in a GLG drawing. There are several types of text objects, which differ both in their behavior and in the number of control points:

- **Fixed Text** displays a string at a specified position. To create a fixed text object, select the *Fixed Text* icon , click in the drawing to define the text's position, then type the string in the text entry box. The text may be scaled when the drawing is zoomed or resized if requested by the setting of the *TextScaling* attribute. The *FontSize* attribute defines the base font and the *MinFontSize* attribute specifies the minimum font size.
- **Fit To Box Text** fits a string within a bounding box. To create a scaled text object, select the *Fit To Box* icon , click on two points to define the diagonal corners of the bounding box, then type the string in the text entry box. Resizing the bounding box changes the font size of the string. The *FontSize* attribute defines the maximum and the *MinFontSize* attribute controls the minimum size allowed. The actual font size is determined dynamically by constraining the text to the rectangle.
- **Wrapped or Truncated Text** fits the string in a bounding box by either wrapping or truncating long text lines that do not fit. The text of this type can also be created by using the *Fit To Box* text icon , then selecting a desired text type from the text type menu popup and clicking on two points in the drawing to define the diagonal corners of the bounding box. The text may be scaled when the drawing is zoomed or resized if requested by the setting of the *TextScaling* attribute. The *FontSize* attribute defines the base font and the *MinFontSize* attribute specifies the minimum font size.
- **Spaced Text** displays a string within a bounding box, with flexible orientation. Changing the box repositions the string. To create a Spaced text object, select the *Object, Create, Text, Spaced Text* menu option, click on two points to define a line, then click on a third point to define the height of the bounding box. Finally, type the string in the text entry box. The third control point is used for line positioning when the text object has several lines. Similar to the Fit To Box text, the Spaced text is fit to the area defined by its three control points, with the *FontSize* and *MinFontSize* defining the maximum and minimum font sizes for scaling.

The control points of the text define a position and the boundaries of the text. The text object's attributes define the string that appears in the drawing and the way it is displayed. All text objects have the following attributes (in addition to control points):

- *TextColor* defines the color of the text.
- *TextString* is the text string displayed by the text object. The *TextString* of a multi-line text can be edited only by using the text edit field of the *Attribute Properties* dialog (ellipsis button ).

- *TextType* can be FIXED, FIT TO BOX, TRUNCATED, WRAPPED, WRAPPED & TRUNCATED or SPACED.
- *TextScaling* defines how the text should be scaled when the drawing is zoomed or resized. It is applicable to all text types except the FIT TO BOX and SCALED, which are always scaled to fit to the text box regardless of the attribute setting. The attribute can have the following values:
  - NONE** to disable text scaling and always display the text using the selected *FontSize*.
  - ZOOM** to scale the text when the drawing is zoomed, but not when it is resized.
  - RESIZE** to scale the text when the drawing is resized, but not when it is zoomed.
  - ZOOM & RESIZE** to scale the text on both zooming and resizing.

The *FontSize* attribute defines the base font size for scaling, which is then adjusted up or down to scale the text. The *MinFontSize* attribute controls the minimum font size allowed. Setting *MinFontSize* to -1 may be used for automatic text label decluttering: the text will disappear if the drawing is zoomed out and there is not enough space to draw the text.

For the ZOOM scaling type, the zoom factor of the drawing's viewport is used to increase or decrease the text's font size. For the RESIZE scaling, a ratio of the current width of the viewport to its base width is used to adjust the font size. The base width of a viewport is defined by the viewport's *BaseWidth* attribute. The font size will be increased to scale the text if the current width is greater than the base width, and decreased if the current width is smaller.

Note that the text is scaled by selecting different size fonts from the viewport's font table, not by changing the dimensions of the characters. If the font sizes defined in the font table for a particular font type are not listed in the ascending order, resizing text objects that use this font type may produce unexpected results.

- *Direction* defines whether text is horizontal, vertical or rotated vertical.
- *Anchoring* defines vertical and horizontal text alignments relative to the text object's control point for the fixed text, or within the text bounding box for other text types.
- *LineAnchoring* defines horizontal alignments of text lines of a multi-line text object.
- *FontType* Specifies the type of the font used to draw the text. The available font types are defined in the drawing's font table attribute.
- *FontSize* defines the font size of the fixed text, or the maximum font size to use for fitting other text types. The available font sizes are defined in the drawing's font table attribute.
- *MinFontSize* specifies the minimum font size to use when scaling the text. Setting *MinFontSize* to -1 activates automatic decluttering feature for scaled text: the text will not be drawn if there is not enough space to fit the smallest font (of size 0) into the text area
- *Text Box* is an optional *Box Attributes* object that controls attributes of an optional box drawn around a text object. A filled box may be used to provide a background for drawing a text object. The text box can be added to an object using the *Add Text Box*  toolbar icon and is also accessible by using the *Add/Edit Text Box* button in the *Object Properties* dialog. The *Box Attributes* object has attributes similar to the polygon object, with a new *Box Offset* attribute that defines additional X and Y offsets between the text and the box's edges. To delete box attributes, use the *Delete Text Box* button in the *Box Attributes*'s properties.

## Marker Object

The marker object is used to mark a position in space. It has a fixed size, and does not change when a window is resized.

To create a marker, click in the marker icon  and define the marker's control point. A marker object has the following attributes:

- *MarkerType* defines the shape of the marker, which can be any combination of the following components: square and filled square, circle and filled circle, diamond and filled diamond, cross and dot.
- *MarkerSize* defines the size of the marker in pixels.
- *FillColor* and *EdgeColor* define colors used to draw the marker's components.

## Spline Object

A **spline** is used to render smooth curves whose shape is controlled by the control points. There are two types of splines:

- *B spline*, which is smoother
- *C spline*, which is not so smooth but passes through the control points.

To create a spline, click on the spline  or filled spline  icon, then click in the drawing area to define control points. The spline inherits the usual polygon attributes, like *LineStyle* and *FillColor*, and has two additional attributes:

- *SplineType* defines the type of a spline, B or C.
- *SplineResolution* is the number of line segments used to render each spline segment. The default value for this is 10. A larger value may be used to increase the rendering quality of splines with large segments or high curvature.

## Connector Object

The connector object may be used to connect other objects in the drawing. It is useful when connecting objects in a diagram drawing.

There are two types of connectors:

- A *recta-linear* connector connects objects with linear segments, maintaining right angles between adjacent segments. It has a variable number of control points that define its segments.
- an *arc* connector connects objects with an arc path and three control points.

To create a recta-linear connector, select a horizontal  or vertical  recta-linear connector icons and define control points; press the *Escape* key to finish. To create an arc connector, click on the arc connector icon  and define 3 control points.

A connector object inherits polygon attributes (*EdgeColor*, *LineWidth* and *LineStyle*) but also has the following attributes:

- *EdgeType* is a read-only attribute defining the type of the connector: *recta-linear* or *arc*.
- *Direction* defines the orientation of the first segment of a recta-linear connector: vertical or horizontal.

## GIS Object

The GIS object is used to display maps in the drawing background. To create a GIS object, click on the GIS Object  icon and select two control points that define the area of the map.

The GIS Object provides the following attributes to control projection, center and extent of the map:

- *FillColor* defines the color of the map's background, which is visible when the map is sufficiently zoomed out.
- *GISDisabled* disables the GIS Object for quick editing.
- *GISProjection* Defines the projection used to render the map: RECTANGULAR or ORTHOGRAPHIC. The rectangular projection displays the world as a rectangular region; the orthographic projection maps the whole world onto a sphere, and is often used for the top-level globe-like views.
- *GISCenter* defines the latitude and longitude of the map to be displayed in the center of the GIS Object. This attribute is of the geometrical (G) type and is a set of three values. The first two values supply the longitude and latitude correspondingly, while the third value must be set to zero.
- *GISExtent* defines the extent of the map visible in the GIS Object. This attribute is of the geometrical (G) type and is a set of three values. The first two values supply the X and Y extents, while the third value must be set to zero.
- *GISAngle* defines the map rotation angle in degrees for the rectangular projection.
- *GISStretch* defines the stretch mode. If the attribute is set to YES, the map is stretched, otherwise the aspect ratio of the map is preserved.
- *GISDataFile* specifies a Server Data File (.sdf) which describes the dataset to be used by the Map Server to generate the map image.
- *GISMapServerURL* specifies the Map Server URL to be used by the Java, C# and JavaScript versions of the Toolkit.
- *GISLayers* defines a list of layers to be displayed in the generated image. The value of this attribute is a comma separated list of layer names, defined in the Server Data File. The "default" string may be used to display the default layers.
- *GISVerbosity* controls diagnostic output. May be set to a value from 0 (no debugging output) to 10 (maximum debugging output) to assist debugging of the Map Server setup.

Refer to the the *GIS Object* section of the GLG User's Guide and Builder Reference Manual for a detailed description of the GIS Object attributes and usage.

## Rendering Attributes

The Rendering object may be attached to most of the graphical primitives to keep an extended set of optional rendering attributes, such as gradient fill, cast shadows, arrowheads and fill dynamics.

The rendering object can be attached to an object by clicking on the Add Gradient  toolbar icon and is also accessible using the *Add/Edit Rendering* button in the *Properties* dialog. It has the following attributes:

- *GradientType* determines both the gradient geometry as well as the colors used for rendering. The following types of the gradient geometry are supported: LINEAR, SPHERICAL, ELLIPTICAL, CONICAL and LINE WIDTH.

The gradient color usage can be DIRECT (the color changing from object color to gradient color) or INVERSED (from gradient color to object color). The LINEAR gradient can also be ACYCLIC (the color changing from the first color to the second color) or CYCLIC (from the first color to the second color and back to the first one). If the value of the gradient type is NONE, the gradient is disabled.

- *GradientColor* defines the second color for the gradient fill. The gradient renders the object with a color smoothly changing from the object's color to the gradient color, or vice versa for the inversed gradient types.
- *GradientAngle* defines the gradient angle for linear gradient or the start angle for conical gradient. The angle is measured counter-clock wise relative to the X axis.
- *GradientLength* defines the relative length of the gradient in the range of 0 to 1, controlling the percentage of the object to be rendered with a gradient fill (the rest of the object will be rendered with a solid color). For example, a value of 1 results in the whole object being rendered with a gradient fill. If the value is equal to 0.5, half of the object will be rendered with a gradient fill, and the other half will be filled with a solid color. If the *Gradient Length* is larger than 1, the gradient fill will extend beyond the boundaries of the object and will be clipped.
- *GradientCenter* defines the center of the gradient fill in relative coordinates, so that the [0;1] range of each of its coordinates corresponds to the object's boundaries in the specified direction. The Z coordinate of the center is ignored. For example, a value of (0.5, 0.5, 0.5) centers the gradient inside the object. A value of (0, 0, 0) positions the gradient's center in the lower left corner of the object.
- *GradientResolution* controls the number of segments used to render the gradient fill. It is ignored for linear gradients in the OpenGL, GTK, Java, C# and JavaScript versions of the Toolkit, and also for the LINE WIDTH gradient.
- *ShadowOffset* specifies the cast shadow offset in pixels. The X and Y coordinates of the offset value define the shadow pixel offset in the corresponding direction. Negative values may be used to inverse the offset's direction. If the value is (0,0), no shadow is rendered.

The Z component of the offset value is interpreted as the shadow transparency. If the Z value is between 0 and 1, the shadow is rendered as transparent, with a greater transparency for the Z values closer to 0.

- *ShadowColor* defines the color of the cast shadow.
- *FillDirection* is the angle that defines the direction of the fill dynamics. The angle is measured

counter-clock wise, relative to the X axis. One of the predefined values (UP, DOWN, LEFT, RIGHT) may be selected from the menu, and ellipsis button  for the *FillDirection* property may be used to enter the angle value in the *Attribute Properties* dialog.

- *FillAmount* controls fill dynamics. It is a relative value in the range of 0 to 1 defining the percentage of the object filled with the object's *FillColor*. The *FillDirection* attribute defines the fill direction. If the value is 1, the whole object is drawn. If the value is less than 1, only a portion of the object's fill will be drawn, as defined by the fill amount. The object's fill type must have FILL enabled in order to be rendered. The object's edge is not affected by the value of the Fill Amount. For text objects, only the text is clipped; the text box is not affected.
- *ArrowType* controls the type of arrow(s) attached to a polygon or any of its subclasses (spline, arc or connector). The arrow type is a composite attribute that determines both the arrows' type (EDGE, FILL or FILL & EDGE) and position (START, END, START AND END or MIDDLE). For arrows positioned in the middle of the polygon, the arrow type also determines an arrow's direction: DIRECT (from start to end point) or INVERSED (pointing from end to start). If the value is NONE, no arrows are rendered. The difference between FILL and FILL & EDGE arrow types becomes visible for lines with line width greater than 1. The geometry of the arrowheads is controlled by the *ArrowShape* attribute and is adjusted to match the line width for thick lines.
- *ArrowShape* controls the shape of the arrow in pixels. The X and Y coordinates of the arrow shape value are used. The X coordinate defines the length of the arrow along the line, the Y coordinate defines the width of the arrow's one side in the direction perpendicular to the line. If null value is specified (0,0,0), the default values are used.

To delete rendering attributes, use the *Delete Rendering* button in the Rendering Properties dialog.

### ***Drawing and Viewport Attributes***

A GLG drawing is contained in a viewport object, which controls the attributes of the drawing's window. The viewport object is also used as a container for dials, meters and other widgets.

The attributes of a drawing window are configured by editing the attributes of the drawing's viewport. To edit the drawing's attribute, press the *Escape* key to unselect any previously selected object, then bring the *Properties* dialog. It displays the following attributes:

- *FillColor* controls the color of the drawing area.
- *EdgeColor* defines the color of the drawing border. On Windows, the border is always black and the attribute is ignored.
- *LineWidth* defines the border width in pixels. On Windows, it may be only 0 or 1.
- *ShadowWidth* specifies the width of the shadow drawn along the viewport's border. The sign of the *ShadowWidth* controls the type of the bevels: raised shadows for positive values and depressed shadows for negative values.
- *ZoomEnabled* enables or disables the run time use of keyboard accelerators for integrated panning and zooming. Refer to the page page 90 of the *GLG User's Guide and Builder Reference Manual* for the list of supported accelerators.
- *ProcessMouse* controls run time mouse events. *Click* events must be enabled for selecting objects with the mouse click. *Move* events must be enabled for processing object tooltips and

*MouseOver* events.

- *DisableInput* may be used to disable input for viewports containing dial and meter widgets.
- *BaseWidth* specifies a viewport's base width used to scale text objects with the RESIZE scaling displayed in the viewport. If the viewport is resized and its current width becomes greater than the base width, the font size of the text objects will be increased. If the current width becomes smaller than the base width, the font size will be decreased.
- *Set BaseWidth* button sets *BaseWidth* of the viewport to its current width.

The *Screen Attributes* button in the viewport's *Properties* dialog display the following additional viewport attributes:

- *OpenGLHint* specifies the driver that will be used for viewport's rendering: OpenGL or GDI. This attribute is ignored in the GTK version of the Toolkit on Linux. In the GTK version of the Toolkit on Linux, the Cairo driver handles anti-aliasing and alpha-blending and the use of OpenGL is disabled by default regardless of the *OpenGLHint* setting. The *ForceOpenGL* attribute may be used to enable OpenGL for viewports that display 3D objects that need hidden surface removal.
- *Stretch XY* enables or disables stretching of the drawing. If set to NO, the drawing preserves the X/Y ratio of objects.
- *PushIn* controls which parts of the viewport are visible when *Stretch XY* is turned off. If set to NO, the best fit is used to map the drawing extent to the window and some parts of the drawing may be invisible. If set to YES, the complete drawing extent will be visible.
- *FonttableFile* specifies a font table file to be used as the default font table. The file is saved using the *Save Font Table* button in the *Fonttable Properties* dialog.
- *Add/Edit Font Table* button provides an access to the properties of the font table which define fonts available for the drawing. A default set of fonts is used by default. Refer to the *Font Table* section on page 180 of the *GLG User's Guide and Builder Reference Manual* for details of the font table attributes and editing.

The *Edit, Keep Edit Ratio* menu option provides access to the *Keep Edit Ratio* attribute of the top drawing viewport. If checked, the drawing will maintain its X/Y ratio to prevent shape distortion of circles and other objects.

## ***Integrated Zooming and Panning***

In the Configurator, the vertical and horizontal scrollbars on the sides of the Drawing Area can be used to scroll the drawing. The drawing can also be scrolled by *Ctrl+clicking* and dragging it with the mouse. To start dragging, the click should happen in an empty area of the drawing. If the whole drawing is completely occupied by objects, the dragging may be started by selecting the *Scroll by Dragging* option of the *View* menu, then clicking anywhere in the drawing and dragging the mouse.

If the drawing needs to be scrolled at run time, the integrated zooming and panning features of the drawing can be used. To activate the run-time scrolling, use the *Edit, Run Time Scrollbars* and select a desired scrolling direction: X, Y or both.

## Adding Geometrical Dynamics

You can add dynamic behavior by adding *move*, *scale* and *rotate* dynamics to graphical objects. To add dynamics to the selected object, click on the *Add Dynamics*  toolbar icon and select one of dynamics types:

- *Move* moves the object in an arbitrary direction by the specified X, Y and Z distances.
- *MoveX* moves the object in the X direction.
- *MoveY* moves the object in the Y direction.
- *Scale* scales the object by the specified scale factor.
- *ScaleX* scales the object in the X direction.
- *ScaleY* scales the object in the Y direction.
- *Rotate* rotates the object by the specified angle around the specified center.

To finish creating the *Move* dynamics, define two points in the drawing to specify the move distance. The *scale* and *rotate* dynamics require one control point to define the center of scaling or rotation. The *MoveX* and *MoveY* dynamics do not require additional points.

When dynamics are added to an object, the *Object Dynamics* dialog displays the dynamics' attributes. The control points of the dynamics are also displayed in the drawing using round markers and may be moved or edited the same way as any other control points. When the object is moved, the dynamic's control points are moved with the object (by default).

The attributes of geometrical dynamics include the following properties:

- *InputValue* is the data value that animates the dynamics.
- *RangeHigh* and *RangeLow* define the high and low ranges of the input value.
- Properties that depend on the type of dynamics, such as:
  - *PointStart* and *PointEnd* for the *Move* dynamics
  - *XDistance* and *YDistance* for the *MoveX* and *MoveY* dynamics
  - *Center*, *ScaleMin* and *ScaleMax* for the *Scale* dynamics
  - *Center*, *AngleStart* and *AngleEnd* for the *Rotate* dynamics.

The input value range defined by the *RangeLow* to *RangeHigh* parameters is mapped to the range of the start and end parameters of the geometrical dynamics. For example, for the *Rotation* dynamics, changing *InputValue* from *RangeLow* to *RangeHigh* rotates the object from *AngleStart* to *AngleEnd*. If the *RangeLow*=0, *RangeHigh*=100, *AngleStart*=0 and *AngleEnd*=90, setting *InputValue*=50 rotates the object by 45 degrees.

To supply dynamic data, a tag may be attached to *InputValue* as described in the *Adding Tags for Data Supply* section.

If an object has geometrical dynamics attached, the *Edit Dynamics*  toolbar icon and the *Dynamics* button in the *Status* panel become active and can be used to access the dynamics' properties.

If more than one geometrical dynamics is added to an object, the dynamics added last will appear at the top of the list. The order of dynamic actions is important and affects the combined dynamic behavior. The *Up* and *Down* buttons in the *Object Dynamics* dialog can be used to change the position of the selected dynamics in the list.

To delete the dynamics, use the *Delete* button in the *Object Dynamics* dialog.

The types of available dynamics may be limited or extended by an OEM vendor, who may also change the name of the dynamic's properties.

### ***Adding Geometrical Dynamics to Control Points***

Geometrical dynamics can be added not only to graphical objects but also to an object's control points. To add dynamics to an object's control point, select the object, then *Shift-click* on the control point to display its *Control Point* dialog. Click on the *Add Dynamics* button and follow the steps for adding geometrical dynamics described above.

To edit a control point's dynamics, use the *Edit Dynamics* button of the *Control Point* dialog.

### ***Adding Attribute Dynamics***

In addition to geometrical dynamics, there are attribute dynamics, such as color dynamics, blinking, text dynamics, which are attached to the object attributes. For example, the *Color List* dynamics may be added to a polygon's *FillColor* attribute to change the color depending on the input value. The *Blinking Alert* dynamics may be attached to an object's *Visibility* attribute to start blinking when the controlling input value goes out of range.

The type of the dynamic behavior is defined not only by the dynamics' type but also by the type of the attribute it is attached to. For example, the *Color Blinking* dynamics can be attached to an object's color to blink by alternating the color value, while the *Blinking* dynamics attached to the object's *Visibility* attribute blinks the object by alternating its visibility.

Attribute dynamics allow the user to implement a variety of dynamic behavior. For example, the text dynamics may be implemented by attaching the *List* dynamics to the text object's *TextString* attribute to change the displayed string depending on the input value. The *Value Display* dynamics may also be attached to the *TextString* attribute to display a numerical value of the input variable. The *Visibility Threshold* dynamics may be attached to the object's *Visibility* attribute to control the object's visibility based on the input value.

To add attribute dynamics to an object, display the object's *Properties* dialog and click on the ellipsis  button next to the attribute you want to add dynamics to. This will display the *Attribute Properties* dialog. Click on the *Add Dynamics* button and select the type of dynamics. The *Attribute Dynamics* dialog will show properties of the added dynamics.

The *Attribute Dynamics* dialog displays parameters according to the type of dynamics, as described below. The parameter of attribute dynamics used for animation, such as *Input Value*, has a tag attached to it, which is used to supply dynamic data as described in the *Adding Tags for Data Supply* section.

Any attribute dynamics attached to an object is visible in the object's *Properties* dialog as a button with the "X" label to the right of the attribute row. Clicking on the button displays a dialog for editing the dynamics.

The *Copy* button in the *Attribute Dynamics* dialog may be used to store the dynamics in the HMI Configurator's clipboard. It may be reused using the *Paste Dynamics* option in the menu which is displayed when the *Add Dynamics* button is pressed.

To delete the dynamics, use the *Delete* button in the *Edit Dynamics* dialog.

The following types of attribute dynamics are available (they may be limited or extended by an OEM vendor):

### *Color List*

The *Color List* dynamics uses an integer index to select a color from a list. It may be applied to color attributes and has the following properties:

**ColorN**

Colors to be used, where *N* is a zero-based color index.

**ColorIndex**

The zero-based index controlling which color to use from the list.

### *Color Threshold*

The *Color Threshold* dynamics compares an input scalar value with a list of thresholds and select a corresponding color from a list of colors. It may be applied to color attributes and has the following properties:

**ColorN**

Colors to be used, where *N* is a zero-based color index.

**ThresholdN**

Thresholds to be used, where *N* is a zero-based threshold index. Threshold values must be specified in the increasing order.

**InputValue**

This value is compared with the thresholds and controls which color to use. The color corresponding to the first threshold smaller than the input value is used.

### *Color Blinking*

The *Color Blinking* dynamics alternates an attribute's color between the two specified colors. It may be applied to color attributes and has the following properties:

**Enabled**

Enables or disables blinking. When set to 0, the blinking is disabled and the *OffColor* is used.

**Interval**

The blinking interval in seconds.

**OnColor**

The first color.

**OffColor**

The second color.

*Color Alert*

The *Color Alert* dynamics changes an attribute's color when the monitored value goes out of the specified range. It may be applied to color attributes and has the following properties:

**ActivateOnEqual**

If set to 1 (*True*), blinking starts when the input value is equal to or exceeds the specified range. If set to 0 (*False*), blinking starts only when the value exceeds the range.

**ColorAlarm**

The color to use when the value goes outside of the *LowLow / HighHigh* range.

**ColorOK**

The default color to use when the value is inside the *Low / High* range.

**ColorWarning**

The color to use when the value goes outside of the *Low / High* range.

**InputValue**

The monitored value.

**RangeHigh**

The *High* range.

**RangeHighHigh**

The *HighHigh* range.

**RangeLow**

The *Low* range.

**RangeLowLow**

The *LowLow* range.

*Color Blinking Alert*

The *Color Blinking Alert* dynamics alternates the attribute's color between the default and alarm colors when the monitored value goes out of the specified range. It may be applied to color attributes and has the following properties:

**ActivateOnEqual**

If set to 1 (*True*), the color is changed when the input value is equal to or exceeds the specified range. If set to 0 (*False*), the color is changed only when the value exceeds the range.

**InputValue**

The monitored value.

**Interval**

The blinking interval in seconds.

**OnColor**

The first color.

**OffColor**

The second color.

**RangeHigh**

The *High* range.

**RangeLow**

The *Low* range.

*List*

The *List* dynamics uses an integer index to select an attribute value from a list. It may be applied to attributes of either D (double, or numerical value) or S (string) type and has the following properties:

**ListIndex**

The zero-based index controlling which attribute value to use.

**ValueN or TextStringN**

Attribute values or text strings to be used, where *N* is a zero-based value or string index.

*Flow*

The *Flow* dynamics may be attached to the *LineType* attribute of lines and polygons to visualize flow of gases and liquids through pipes. The dynamics shifts a line type pattern along the length of the line to animate the flow, It has the following properties:

**DisabledLineType**

The line type to use when the flow is disabled. The default value is 0 (solid line).

**EnabledLineType**

The line type pattern to use when the flow is enabled.

**FlowEnabled**

Enables the flow animation if set to 1, disables it if set to 0.

**FlowInterval**

Controls the flow speed by defining a timer interval (in milliseconds) that is used to update the animation.

**FlowInversed**

Inverses the flow direction if set to 1. The default value is 0 (direct flow).

*Threshold*

The *Threshold* dynamics compares an input scalar value with a list of thresholds and select a corresponding attribute value from a list of values. It may be applied to attributes of either D (double, or numerical value) or S (string) type and has the following properties:

**InputValue**

This value controls which attribute value to use. It is compared with the thresholds in the list and the value corresponding to the first threshold smaller than the input value is used.

**ThresholdN**

Thresholds to be used, where *N* is a zero-based threshold index. The thresholds must be specified in the increasing order.

**ValueN or TextStringN**

Attribute values or text strings to be used, where *N* is a zero-based value or string index.

### *Range Alert*

The *Range Alert* dynamics changes the attribute's value when the monitored value goes out of the specified range. It has the following properties:

**ActivateOnEqual**

If set to 1 (*True*), the attribute's value is changed when the input value is equal to or exceeds the specified range. If set to 0 (*False*), the value is changed only when the input value exceeds the range.

**InputValue**

The monitored value.

**ValueOK**

The value to use when the value is inside the *Low / High* range.

**ValueWarning**

The value to use when the value goes outside of the *Low / High* range.

**ValueAlarm**

The value to use when the value goes outside of the *LowLow / HighHigh* range.

**RangeHighHigh**

The *HighHigh* range.

**RangeHigh**

The *High* range.

**RangeLow**

The *Low* range.

**RangeLowLow**

The *LowLow* range.

The *ValueWarning*, *RangeHighHigh* and *RangeLowLow* properties are not present in the *Range Alert* dynamics attached to the *Visibility* attribute.

### *Blinking*

The *Blinking* dynamics alternates an attribute's value between the two specified values. It may be attached to any attribute of D type (double, or numerical value), including the object's visibility. It has the following properties:

**Enabled**

Enables or disabled blinking. When set to 0, blinking is disabled and the *OffValue* is used.

**Interval**

The blinking interval in seconds.

**OnValue**

The first color.

**OffValue**

The second color.

### *Blinking Alert*

The *Blinking Alert* dynamics alternates the attribute's value when the monitored value goes out of the specified range. It may be applied to any attribute and has the following properties:

**ActivateOnEqual**

If set to 1 (*True*), blinking starts when the input value is equal to or exceeds the specified range. If set to 0 (*False*), blinking starts only when the input value exceeds the range.

**InputValue**

The monitored value.

**Interval**

The blinking interval in seconds.

**OnValue**

The value to use for blinking when the value goes outside of the range.

**OffValue**

The default value to use when the value is inside the range.

**RangeHigh**

The *High* range.

**RangeLow**

The *Low* range.

*Visibility Threshold*

The *Visibility Threshold* dynamics compares an input scalar value with the specified threshold and set the object visibility to a matching value. It may be applied to an object's *Visibility* attribute and has the following properties:

**InputValue**

This value is compared with the threshold and controls which attribute value to use.

**Threshold**

The threshold.

**VisState0**

The visibility value to use when the input value is less than the threshold. May be set to 0 or 1.

**VisState1**

The visibility value to use when the input value is greater than the threshold. May be set to 0 or 1.

*Value Display*

The *Value Display* dynamics may be attached to the *TextString* attribute of text objects to display a numerical value using the specified format. It has the following properties:

**InputValue**

The value to be displayed.

**Label**

The label used to annotate the value.

**MinLength**

The minimum number of characters used to display the value. If the number of characters is less than *MinLength*, the value is padded with spaces on the left.

**Precision**

The number of digits after the decimal point in the value display.

**Separator**

The string used as a separator between the label and the value.

**Units**

The unit string displayed after the value.

*Text Display*

The *Text Display* dynamics may be attached to the *TextString* attribute of text objects to display a string using the specified format. It has the following properties:

**InputString**

The string to be displayed.

**Label**

The label used to annotate the string value.

**MinLength**

The minimum number of characters used to display the string. If the number of characters is less than *MinLength*, the string is padded with spaces on the left.

**Separator**

The string used as a separator between the label and the string.

**Suffix**

The second annotation displayed after the string.

*JavaScript*

The *JavaScript* dynamics allows using an arbitrary JavaScript expression to generate an output values of the transformation based on values of its input parameters. In the HMI Configurator, the JavaScript dynamics is provided for the D (double) and S (string) attributes. The dynamics has the following properties:

**ArgN**

Input parameters, where N is 1-based index of the input parameter.

**JavaScript**

The JavaScript expression.

Refer to the *JavaScript* section on page 216 of the *GLG User's Guide and Builder Reference Manual* for information on using input parameters in the JavaScript expression and examples of the JavaScript syntax.

*Paste Dynamics*

This option will be present in the menu if a transformation was copied using the *Copy* button of the *Attribute Dynamics* dialog. Using this option adds a copy of the stored dynamics to the attribute.

If the *Treat Dynamics as Global on Copy* option of the *Options, Dynamics Options* menu is set, the parameters of the added dynamics, such as a list of colors, will be constrained to the corresponding parameters of the original dynamics. Changing parameters of either dynamics changes them in all global copies.

The *Treat Dynamics as Global on Copy* option must be set before copying the dynamics, and works until the drawing is reloaded.

## Adding Tags for Data Supply

All dynamics defined in the drawing are driven by changing values of their attributes. For example, if *Move* dynamics is attached to an object, it is animated by changing the dynamics' *InputValue* attribute. If an object has *ColorList* dynamics, it is animated by changing the value of its *ColorIndex* attribute. Since all attributes are inherently dynamic, any object attribute may be animated directly by supplying a new attribute value. For example, the *FillColor* attribute may be set to a new RGB color value.

To simplify drawing animation, tags may be attached to attributes to facilitate connectivity to real-time data. A tag is added to an attribute in a form of a tag object which has several attributes of its own:

- *TagName* is a string attribute that assigns a meaningful name to a tag to identify the tag in an application and assist the user in mapping the tags to data sources using a *Tag Browser*. *TagName* is persistent and does not change when the tag's data source is modified.
- *TagSource* is a string attribute that defines a source of run-time data used to dynamically change the value of the attribute the tag is attached to. For example, *TagSource* may specify a name of a field in the process database used to supply real-time data for animating the attribute. Any attribute with a tag attached may be accessed by the tag's *TagSource*.
- The *TagComment* attribute is a string that may be used to store any additional user-defined information associated with the tag.

Tags simplify data connectivity by providing mapping between dynamic attributes and data sources used to animate the drawing. The tag's *TagSource* attribute specifies the source of dynamic data for each attribute, such as a field in a process database. When an application receives a data change event for a particular field, it updates the corresponding tag.

In the HMI Configurator, the user can browse tags using the *Tag Browser*, which display a list of all tags defined either in the drawing or in the selected object. To change the database mapping, the user can browse tags, select the tag whose database connection needs to be changed and edit its *TagSource*.

### Adding a Tag to an Object Attribute

To add a tag to an object's attribute, select an object and bring its *Properties* dialog. Click on the ellipsis button **...** next to some attribute, for example *LineWidth*, to display the *Attribute Properties* dialog. Click on the *Add Tag* button to add a tag to the attribute: the *Data Tag* dialog will be displayed.

Data Tag Dialog

When a tag is added, its *TagName* and *TagSource* are initially set to the string “unset” and may be edited in the *Data Tag* dialog. Type “*LineWidthAnimation*” in the *TagName* field as a tag description, then enter “*LWValue*” in the *TagSource* field.

Close the *Data Tag* dialog by clicking on the *OK* button.

The values of the tag's *TagName* and *TagSource* attributes are displayed in the *Tag* field of the *Attribute Properties* dialog as two strings separated by the '/' character. To edit the tag again, click on the *Edit Tag* button of the *Attribute Properties* dialog.

The object's *Properties* dialog indicates the presence of added tags by placing a button with the “T” label to the right of the attribute row. Clicking on the button activates the *Data Tag* dialog.

To delete the tag, use the *Delete* button in the *Data Tag* dialog.

### Mapping Tags to DataSources in the HMI Configurator

A tag's *TagSource* is usually set to the name of a field in a process database that provides the input value for animation. When the value of the database field changes, the application supplies the new tag value. Since *TagSources* are global, the application doesn't need to know which object or attribute they are connected to. If several tags have the same *TagSource*, all of them will be updated when the new value for this *TagSource* is supplied at run time.

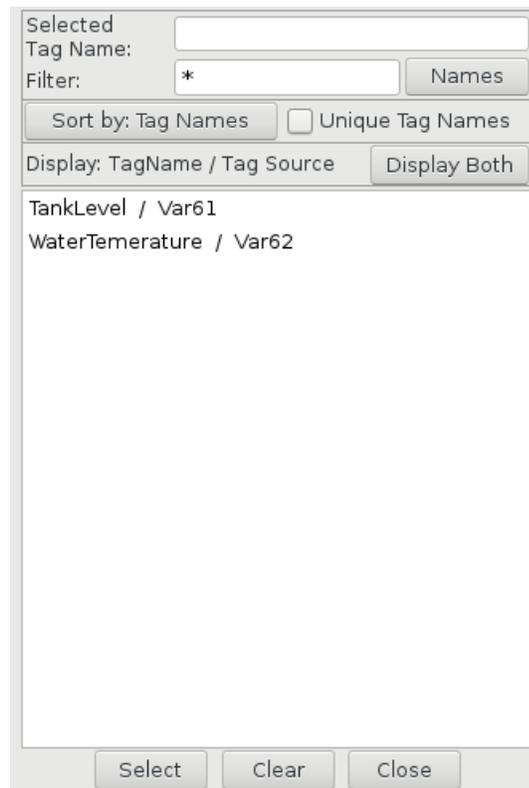
A custom *Data Browser* may be supplied by an OEM vendor to display a list of available tags in the process database. The *Data Browser* is activated by the *Browse* button in the *Data Tag* dialog and displays a list of available tags. When a user selects a tag from the list, it is assigned to the tag's *TagSource* field.

The *Data Browser* uses a custom data browser DLL provided by the OEM vendor. If no custom DLL is provided, a default demo DLL provides an example of browsing hierarchically organized groups of tags.

### Browsing Tags Defined in the Drawing

All tags defined in the drawing can be listed using the *Tag Browser*. Click on the *Tags*  toolbar button or use *Object, Tags* menu option to activate the *Tag Browser*. If no objects are selected, the *Tag Browser* will show all tags defined in the drawing. If an object is selected, the *Tag Browser* will show only tags defined for the object.

Each tag's entry shows its *TagName* and *TagSource* attributes separated by the '/' character. Clicking on a tag entry selects it, showing the *Data Tag* dialog for editing the tag. The *Edit Value* button of the *Data Tag* dialog activates the *Attribute Properties* dialog for editing the value of the attribute the tag is attached to.



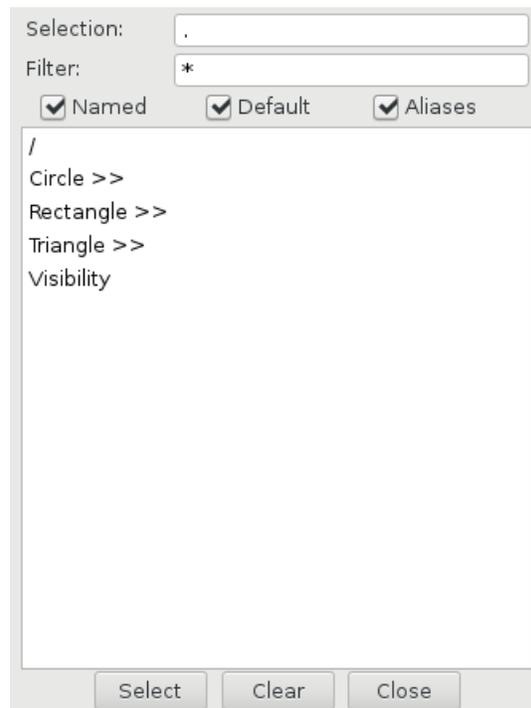
Tag Browser

The *Tag Browser* also contains entries for sorting tags based on various criteria. The *Filter* field may be used to display only a subset of tags matching a regular expression that may contain the ? (any character) and \* (any sequence of characters) wild cards. The *Selected Tag Name/Source* field may be used to select a tag by typing its *TagName* or *TagSource*.

### Using Resources to Browse Objects

An object may be assigned a name (via the *Name* property) to make it accessible via a resource browser. The *Resource Browser* can be used to list all named objects in the drawing and edit their attributes without selecting the objects. Click on the *Resources*  toolbar icon or use *Object,*

*Resources* menu option to display the *Resource Browser*. The following picture shows the *Resource Browser* that displays resources of a group containing objects named *Circle*, *Rectangle* and *Triangle*.



Resource Browser

When no objects are selected, the resource browser lists all named objects contained in the drawing, as well as the attributes of the drawing. Entries that have sub-resources, such as named objects, are annotated with the “>>” suffix in the resource browser. Double-clicking on such entry displays a list of resources inside it. Clicking on a resource entry without “>>” opens a dialog for editing its value. Double-click on the “.” entry to go back to the previous level in the resource hierarchy, one level at a time.

If some object is selected, the resource browser displays the object’s resources, regardless of whether the object is named or not.

Any dynamics added to an object in the HMI Configurator are visible in the resource browser as a resource of that object, which may be used for a fast access to the dynamics’ attributes. For example, the *ColorList* dynamics added to an object’s *FillColor* attribute will appear as the “*FillColor:ListDynamics>>*” resource of the object. Double-clicking on this entry will display parameters of the dynamics.

The *Filter* field may be used to display a subset of resources whose names match a regular expression that may contain the ? (any character) and \* (any sequence of characters) wild cards. The *Selection* field may be used to select a resource entry by typing its name.

The *Resource Browser* may be disabled using the *glg\_hmi\_config* file.

## Prototyping Dynamic Behavior Using Run Mode

The Run mode is used to test the drawing's animation and is started by clicking on the *Start*  toolbar icon. The Run mode toolbar contains controls for pausing the animation and changing its update rate. It also displays performance-related statistics, such as elapsed time and update rate.

The *Run* mode may use a custom data acquisition DLL supplied by an OEM vendor to animate the drawing with real-time data using tags defined in the drawing.

A random data generator may also be used to test the animation via the *\$datagen* command supplied when the Run mode is started. The following sample command animates two tags named *RotateAngle* and *MoveDistance*:

```
$datagen -tag -sin d 0 90 RotateAngle
        -tag d 0 100 MoveDistance
```

The *RotateAngle* tag is animated with sinusoidal data in the range from 0 to 90, while the *MoveDistance* tag is animated with random data in the range from 0 to 100. The “d” parameter specifies the double (numerical) data type.

The drawing can also be animated using resources. For example, the following command animates the *Value* attribute of the dial widget named *Dial1*:

```
$datagen -sin d 0 10 Dial1/Value
```

Refer to the the *The Data Generation Utility* chapter on page 326 of the *GLG Programming Reference Manual* for a complete list of the *datagen* parameters.

Use the *Stop*  toolbar icon to stop the Run mode.

## Defining Alarms

An alarm can be attached to any attribute to monitor its value. The range alarm defines a normal range of the monitored value and generates an alarm message when the value goes outside of the normal range. There is also a change alarm that generates a message every time the monitored value changes. Alarms are defined in the drawing, but alarm messages are processed in the application's code.

To add an alarm to an attribute, click on the ellipsis button  next to the attribute in the *Properties* dialog to bring the *Attribute Properties* dialog, then click on the *Add Alarm* button and select an alarm type. The *Edit Alarm* dialog will be displayed to edit the alarm's parameters.

While alarm attributes depend on the type of alarm as described in the following sections, all alarm objects share the following common attributes:

- *Alarm Label* contains a user-defined label used to identify the alarm.
- *Enabled* parameter enables or disables the alarm by setting its value to 1 or 0, respectively.

The rest of the alarm attributes depend on the alarm type and are described in the following sections.

If an attribute has an alarm attached, the *Add Alarm* button changes to *Edit Alarm* and can be used to edit the alarm. The object's *Properties* dialog indicates the presence of added alarms by placing a button with the "A" label to the right of the attribute row. Clicking on the button activates the *Edit Alarm* dialog.

To delete an alarm, use the *Delete* button in the *Edit Alarm* dialog.

The following alarm types are available:

### *Range Alarm*

The *Range* alarm can be attached to an attribute of a D type (double, or numerical value) to monitor its value. The alarm message is generated when the value goes below or above the specified *High/Low* range.

The *Range* alarm has the following attributes:

#### **High**

Defines the *high* range of the value. The alarm message with the *High* subaction is generated when the value reaches or exceeds the *High* range.

#### **Low**

Defines the *low* range of the value. The alarm message with the *Low* subaction is generated when the value drops below or equal to the *Low* range.

### *Range2 Alarm*

This alarm is similar to the *Range* alarm but adds the following attributes that define a second set of thresholds in addition to *High* and *Low*:

#### **High High**

Defines the "*high high*" range of the value. The alarm message with the *HighHigh* subaction is generated when the value reaches or exceeds the *HighHigh* range.

#### **Low Low**

Defines the "*low low*" range of the value. The alarm message with the *LowLow* subaction is generated when the value drops below or equal to the *LowLow* range.

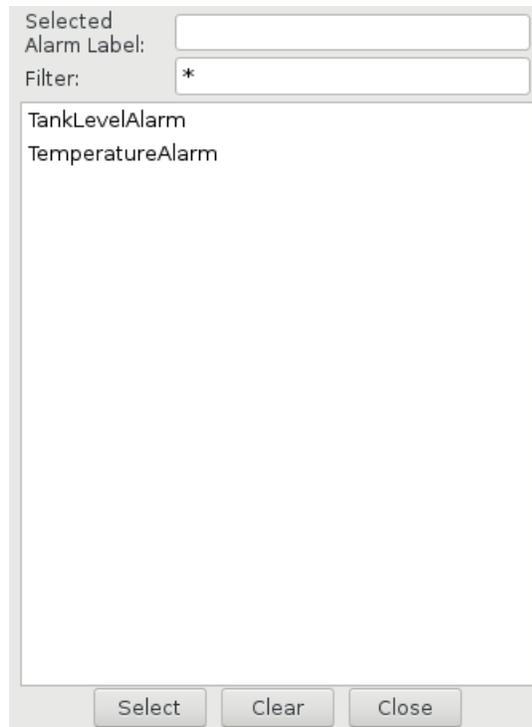
### *Change Alarm*

The *Change* alarm can be attached to an attribute of any type (D, S or G) to monitor changes of its value. The *Change* alarm does not define any new attributes in addition to *Alarm Label* and *Enabled*.

The *Change* alarm generates an alarm message with the *ValueChange* action and NULL subaction every time the value is changed.

### Browsing Alarms

The *Alarm Browser* may be used to display a list of alarms defined in the drawing or in individual objects. Select the *Object, Alarms* menu option to display the *Alarm Browser*.



Alarm Browser

If no object is selected, all alarms defined in the drawing will be displayed. If some object is selected, only the alarms attached to the attributes of that object will be listed. Each alarm entry in the *Alarm Browser* shows the alarm's *AlarmLabel* attribute.

When an alarm entry in the alarm browser is selected with the mouse, its *Alarm Dialog* is displayed. Clicking on the *Prop* button in the dialog opens the *Attribute Properties* dialog to edit the attribute (property) the alarm is attached to.

The *Filter* field may be used to display only a subset of alarms whose *AlarmLabel* matches a regular expression that may contain the ? (any character) and \* (any sequence of characters) wild cards. The *Selection* field may be used to select an alarm by typing its *AlarmLabel*.

## Using GLG Widgets and Palettes

The *Palettes* menu provides access to palettes of pre-built widgets and objects. A widget from a palette can be added to the drawing by simply clicking on the widget in the palette.

By default, the widget will be added and positioned in the center of the drawing. To enable manual widget positioning with the mouse, check the *Palettes, Manual Widget Positioning* menu option. To change the default setting, set the *ManualWidgetPositioning* variable in the *glg\_hmi\_config* configuration file.

The *Palettes, Make Widget Viewports Transparent* menu option may be checked to convert all widgets that use viewports, such as buttons or dials, to use light viewports with transparent background. The *Arrange, Convert Viewport* menu can be used to convert viewports to light viewports and back at a later time.

The widgets have built-in dynamics and are highly configurable. The appearance and behavior of each widget is controlled by widget resources.

Each widget also has public properties that facilitate simplified widget editing. Public properties list the most commonly used widget resources which could be edited via the *Properties*  button. A complete list of widget resources is accessible via the Resource Browser .

A widget may be animated by supplying dynamic data to widget properties, such as the *Value* property of a dial or tank widget. A tag can be added to a widget property to specify the data source for the real-time data. A tag stores the name of the data source, and this name can be used in a program at run time to supply real-time data values. To add a tag, click on the ellipsis  button next to the desired property in the *Properties* dialog, click on *Add Tag* in the activated property dialog and enter a data source name in the *Tag Source* field in the *Tag* dialog.

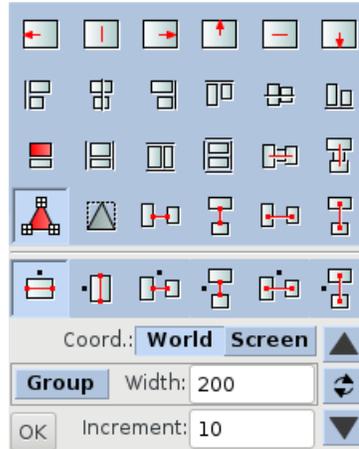
By default, only the *Custom Objects* and *Custom Widget Samples* palettes are installed. Other palettes are optional. Possible palettes include Real-Time Charts, 2D Graphs, 3D Graphs, Controls, Avionics, Process Control Symbols, Electrical and Electronic Circuit Symbols, and Special Widgets palettes, as well as the palettes provided by an OEM vendor.

The *Palettes* menu lists all available palettes. To display the *Custom Widget Samples* palette, select it from the palettes list. To add an object from a palette into the drawing, click on its icon in the palette. For example, click on the tank component in the *Custom Widget Samples* palette to insert a copy of the tank into the drawing. Give it a name and adjust its shape using the resize box.

## **Object Layout and Alignment**

If you need to align or layout a group of objects, *Layout Toolbox* provides a point and click interface for such functionality. Examples of layout and alignment operations include aligning several objects horizontally or vertically, setting the same width for several objects, positioning objects within a defined distance from each other, or distributing the objects evenly across the defined extent in a horizontal or vertical direction.

Click on the *Layout Toolbox*  button in the toolbar to activate a *Layout Toolbox*. Alternatively, the *Layout Toolbox* can be activated using *Layout, Layout Toolbox* option in the main menu. The following picture shows the *Layout Toolbox*:



The top of the *Layout Toolbox* contains icons for actions that do not require a value, such as aligning objects or making them the same width or height. The bottom of the toolbox contains controls for actions that require a value, such as setting an object's width or height to a specified value.

To experiment with the *Layout Toolbox*, create three rectangles using the *Filled Rectangle*  button and position them horizontally, one next to another. Select all rectangles by clicking and dragging the mouse to define a rectangular area that encloses all three rectangles. This will create a temporary group containing all three rectangles.

Click on the *Align Bottom*  button in the *Layout Toolbox* and notice that the bottom of all rectangles will be aligned.

The *Select Anchor Object*  button in the *Layout Toolbox* may be used to define an anchor object. If an anchor is selected, its dimensions are used to align other objects. Click on the *Select Anchor Object*  button, then click on one of the rectangles to define it as an anchor.

If the anchor is not defined, one of the objects in the group will be chosen as an anchor automatically. Once the anchor is defined, it will be persistent for all layout operations until another anchor is chosen or the objects are unselected.

Click on the *Set Same Height*  button to set the height of all rectangles to the height of the anchor rectangle.

The icons in the upper half of the toolbox perform actions when you click on them, while the icons in the lower half of the dialog define actions to be performed when a numerical value is entered.

For example, follow these steps to position the rectangles with an even horizontal space between them:

- Click on the *Set Horizontal Space*  button to select the operation to be performed. Notice that the label of the text edit box at the bottom of the *Layout Toolbox* displays the selected

operation and is set to *Horiz.Space*.

- Enter 50 as the value in the *Horiz. Space* text box and press *Enter*. This will position all rectangles to be within 50 world coordinates from each other in the horizontal direction.
- Use the Up and Down buttons to increase or decrease the space between the rectangles using the specified *Increment*.

The *Coord: World / Screen* control toggles between world and screen coordinates.

Press the *Escape* key to unselect the rectangles.

## Defining Object Tooltips

The HMI Configurator provides support for integrated object tooltips. To add a tooltip to an object, select the object and use *Object, Object Tooltip, Add Tooltip*. The *Object Tooltip* dialog will appear. Enter the tooltip string into the *Value* field and close the dialog.

To enable tooltips in the drawing, the *ProcessMouse* attribute of the drawing has to be set to *Move & Click*. Press the *Escape* key to unselect the object, then bring the *Properties* dialog: it will display attributes of the drawing's viewport. Set *ProcessMouse* to *Move & Click*.

The HMI Configurator's Run mode can be used to test the tooltips. Press the *Start*  toolbar icon to start the Run mode, move the mouse over the object and hold it until the tooltip appears. Use the *Stop*  toolbar icon to stop the Run mode.

The tooltip can be edited or deleted using the *Edit Tooltip* and *Delete Tooltip* entries in the *Object, Object Tooltip* menu.



## Chapter 2

## GLG HMI Configurator Menus

# 2

This chapter presents descriptions of all the options of the GLG HMI Configurator's menus. They are organized by menu name, in the same order as they appear in the HMI Configurator's menu bar:

- The *File* Menu provides options that mainly deal with saving and printing drawing files, saving images and various exported files. These options are described starting on page 51.
- The *Palettes* Menu provides access to palettes of widgets and other pre-built objects. These options are described starting on page 57.
- The *Edit* Menu provides options for selecting, editing and copying objects. These options are described starting on page 59.
- The *Undo* Menu provides options for undoing editing changes. These options are described starting on page 62.
- The *View* Menu provides options for controlling your view of the drawing. These options are described starting on page 63.
- The *Arrange* Menu provides options for working with groups, as well as other miscellaneous options described starting on page 64.
- The *Layout* menu provides options for aligning and positioning objects. These options are described starting on page 68.
- The *Object* Menu provides options for creating and manipulating objects. These options are described starting on page 73.
- The *Run* Menu provides options testing the drawing animation. These options are described starting on page 80.
- The *Options* Menu provides options that control the appearance and function of the GLG HMI Configurator. These options are described starting on page 81.

The toolbar below the menu bar provides convenient shortcuts for accessing the most often used menu choices. To see the tooltip showing the function of a toolbar's button, hold the mouse inside the button until the tooltip appears.

### **File**

The *File* menu provides options for creating, loading and saving drawing files, exporting/importing strings and tags, printing drawings, generating images, and closing the GLG HMI Configurator.

### ***New Drawing (Resize and Stretch)***

*New Drawing (Resize and Stretch)* creates a new resizable drawing. When the drawing is resized, all objects in the drawing are resized as well. If the window is resized in such a way that its width/height ratio is not maintained, the drawing is stretched to fit the new window size, which makes objects such as circles appear stretched.

Several options are provided to target screens with different aspect ratios:

- 1:1 Width/Height Ratio*
- 4:3 Width/Height Ratio*
- 16:9 Width/Height Ratio*
- Custom Width/Height Ratio*
- Width/Height Ratio From Config File*

The first three options create a resizable widget with the selected width/height ratio.

The *Custom Width/Height Ratio* option may be used to create a resizable widget with a custom ratio that will be defined by the ratio of the *SpanX* and *SpanY* values of the widget's viewport. These values are entered in the popup dialog and define the widget viewport's drawing span.

The last option uses the aspect ratio defined by the *CustomSpanX* and *CustomSpanY* parameters in the *glg\_hmi\_config* file.

### ***New Drawing (Resize, No Stretch)***

*New Drawing (Resize, No Stretch)* creates a new resizable drawing. When the drawing is resized, all objects in the drawing are resized as well. The drawing maintain its X/Y ratio, so that objects such as circles always maintain their circular shape.

Several options are provided to target screens with different aspect ratios:

- 1:1 Width/Height Ratio*
- 4:3 Width/Height Ratio*
- 16:9 Width/Height Ratio*
- Width/Height Ratio From Config File*

The last option uses the aspect ratio defined by the *CustomSpanX* and *CustomSpanY* parameters in the *glg\_hmi\_config* file.

### ***New Drawing (Fixed Scale, No Stretch)***

*New Drawing (Fixed Scale, No Stretch)* creates a non-resizable drawing where objects do not change their size when the drawing is resized. Instead, more or less of the drawing area is shown in the window. Since the drawing is not stretched, the objects in the drawing always keep their X/Y ratio.

The following options are provided:

- Custom Size*
  - allows creating a fixed scale drawing of an arbitrary size specified via a popup dialog.
- Size From Config File*
  - creates a fixed scale viewport with the size specified in the *glg\_hmi\_config* file.

## ***Reset Drawing***

*Reset Drawing* re-initializes the current drawing.

For composite objects such as graphs, *Reset Drawing* erases volatile resources such as graph's data.

## ***Work Area Size***

The *Work Area Size* submenu provides options for setting work area size. If the size is set, the drawing assumes the work area size when the widget's viewport is entered to edit its drawing. The work area size is preserved when the HMI Editor is resized.

### *Unset*

*Unset* unsets the previously set work area size. The current size of the Drawing Area will be used as the work area.

### *Custom Size*

*Custom Size* sets the work area size to the values entered in the popup dialog.

### *Size From Config File*

*Size From Config File* sets the work area size to the size specified in the *glg\_hmi\_config* file.

## ***Adjust \$Widget Size***

*Adjust \$Widget Size* submenu provides options for resizing the top *\$Widget* viewport of the drawing.

### *To Visible Area*

*To Visible Area* adjusts dimensions of the *\$Widget* viewport to take all space available in the Drawing Area while maintaining its internal width/height ratio defined by the viewport's SpanX and SpanY attributes.

### *To Work Area Size*

*To Work Area Size* adjusts dimensions of the *\$Widget* viewport to the size of the work area. This option is active only if the work area size has been set.

### *Set Custom Size (Pixels)*

*Set Custom Size (Pixels)* sets the width and height of the *\$Widget* viewport to the values entered in the popup dialog.

## ***Resize Drawing***

Changes the size of a fixed scale drawing.

## ***Open***

*Open* loads a drawing from a file.

When you select *Open*, the HMI Configurator prompts you to select a file name, using the standard file selection dialog.

If a drawing is already open, *Open* discards the current drawing. The HMI Configurator asks if you want to discard the current drawing, but it does not prompt you to save the changes.

## ***Open URL***

*Open URL* loads a drawing from a URL.

In the Linux/Unix environment, the `GLG_WGET_PATH` environment variable must be set to point to the `wget` utility executable to enable this option.

## ***Load Template***

This option is similar to the *Open* option for loading a drawing. However, *Load Template* doesn't allow the user to save the modified drawing back to the same file, to avoid accidentally overwriting the drawing used as a template for creating other drawings.

## ***Recent Drawings***

*Recent Drawings* displays a list of the recently edited drawings. Select one of the recent drawings from the list to load it.

If a drawing is already open, loading a recent drawing discards the current drawing. The HMI Configurator asks if you want to discard the current drawing, but it does not prompt you to save the changes.

## ***Save***

*Save* saves the current drawing to a file.

The first time you save a drawing, the HMI Configurator prompts you for a file name, using the standard file selection dialog. If the drawing has been saved before, it asks if you want to overwrite the existing file.

The HMI Configurator can save files in two different formats:

- Binary, which loads quickly but is not portable across platforms that use different binary data representations.
- ASCII, which is completely portable across platforms but loads slower than binary.

There is also an option for saving compressed drawings. Compressed drawings are smaller but load slower than non-compressed.

The default format for saving drawings is compressed ASCII. To change the format, use *Options*, *Save Format* and *Save Compressed*; see page 82.

### ***SaveAs***

*Save As* saves the current drawing to a different file.

When you select *Save As*, the HMI Configurator prompts you to enter a file name, using the standard file selection dialog. The usual file name extension for a GLG drawing is *.g* though it is not required.

The default format for saving drawings is compressed ASCII. To change the format, use *Options*, *Save Format* and *Save Compressed*; see page 82.

### ***Print (Windows and GTK on Linux only)***

*Print* sends the current drawing to the printer, using the standard system print facilities. Use *Export PostScript* to save a PostScript image of the drawing to a file.

*Print* uses the print configuration set by the *File*, *Print Settings* options.

### ***Export PostScript***

*Export PostScript* saves a PostScript image of the current drawing (in its current state) to a file.

*Export PostScript* uses the print configuration set by the *File*, *Print Settings* options.

### ***Print Settings***

The *Print Settings* submenu provides options to let you set up the printer and the PostScript output.

#### ***Page Layout***

*Page Layout* specifies how to map the drawing to the printed page.

*Page Layout* presents a viewport that corresponds to the printed area. The position and size of the viewport relative to the *Drawing Area* define the position and size of the area in which the drawing will be printed relative to the page.

Resize the page layout viewport with the mouse to define the printing area. Delete the viewport when you have finished.

This option applies to both printing and exporting PostScript.

### *Stretch*

*Stretch* prints the drawing using the full area of the page. The drawing is scaled to fill the printing area, so the proportions of the printed drawing may not correspond to the drawing's actual proportions. Using the appropriate orientation (portrait or landscape) can help reduce distortion; to preserve the drawing ratio, turn *Stretch* off.

This option applies to both printing and exporting PostScript.

### *Center*

*Center* causes the printed output to be centered on the page. If turned off, the printed output will start at the upper left corner.

This option applies to both printing and exporting PostScript.

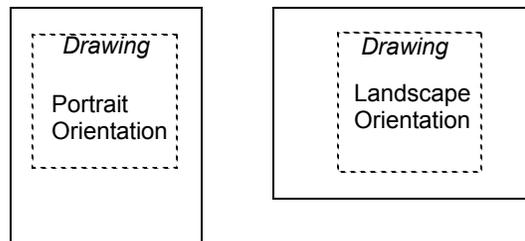
### *PostScript Level*

*PostScript Level* specifies which version of PostScript is generated. Level 3 is required for proper PostScript output of images with transparent background.

### *PostScript Orientation*

*PostScript Orientation* determines whether the drawing is printed in portrait mode (across the width of the page) or in landscape mode (down the height of the page).

For printing, the page orientation is set in the *Print* dialogWindows.



**Portrait and Landscape Orientations**

### *Save Image*

*Save Image* saves the image of the visible part of the drawing area into a file in either the JPEG or PNG format. The format of the image is defined by the *SaveImageFormat* variable in the *glg\_hmi\_config* file.

### *Save Image Full*

*Save Image Full* saves an unclipped image of the whole drawing into a file in either the JPEG or PNG format. The format of the image is defined by the *SaveImageFormat* variable in the *glg\_hmi\_config* file.

## ***Export Strings***

*Export Strings* exports all strings defined in the drawing into a file.

Refer to the *Localization Support* chapter on page 255 of the *GLG User's Guide and Builder Reference Manual* for information about the string translation file format.

## ***Import Strings***

*Import Strings* imports strings from a string translation file, replacing matching strings in the drawing.

Refer to the *Localization Support* chapter on page 255 of the *GLG User's Guide and Builder Reference Manual* for information about the string translation file format.

## ***Export Tags***

*Export Tags* exports all tag names defined in the drawing into a file.

Refer to the *Tag Export and Import Features for Run-Time Tag Mapping* chapter on page 259 of the *GLG User's Guide and Builder Reference Manual* for information about the tag file format.

## ***Import Tags***

*Import Tags* imports tag names from a tag translation file, replacing matching tag names in the drawing.

Refer to the *Tag Export and Import Features for Run-Time Tag Mapping* chapter on page 259 of the *GLG User's Guide and Builder Reference Manual* for information about the tag file format.

## ***Exit***

*Exit* closes the GLG HMI Configurator.

If a drawing is already open, the HMI Configurator asks if you want to discard the current drawing, but it does not prompt you to save the changes. You must explicitly save the drawing using *File, Save* before exiting the GLG HMI Configurator.

## ***Palettes***

The *Palettes* menu contains the *Shapes, Images and Text* palette used for creating graphical primitives, such as polygons, arcs, images and text objects, and also provides access to palettes of widgets and other pre-built objects. By default, only the *Custom Objects* and *Custom Widget Samples* palettes are installed. Other palettes are optional. Possible palettes include *Real-Time Charts, 2D Graphs, 3D Graphs, Controls, Avionics, Process Control Symbols, Electrical and Electronic Circuit Symbols*, and *Special Widgets* palettes, as well as the palettes provided by an OEM vendor.

The *Palettes* Menu lists all available palettes of pre-built objects. To display a palette, select it from the palettes list. To add an object from a palette into the drawing, click on its icon in the palette. The HMI Configurator will insert a copy of the object into the drawing. Give the object a name and adjust its shape using the resize box.

When no optional palettes are installed, the *Palette* Menu provides the *Custom Objects*, *Custom Widget Samples*, *Read Palette* and *Read Directory* options.

### ***Manual Widget Positioning***

*Manual Widget Positioning* controls how widgets from the widget palettes are positioned in the drawing. If the toggle is checked, selecting a widget from the palette and clicking in the drawing will insert the widget at the position of the mouse click. If the option is unchecked, the widget will be positioned in the center of the drawing area.

### ***Adjust Widget Size***

*Adjust Widget Size* controls adjusting widget sizes when widgets from the widget palettes are inserted into the drawing. If the toggle is **checked** (default), the size of the inserted widget is adjusted to preserve the same initial size in screen pixels regardless of the size of the drawing or the extent of the viewport the widget is placed into. The widget can then be resized as needed.

If the toggle is **unchecked**, the widget is inserted using its original size in world coordinates without any size adjustments. The widget may appear very small when it is inserted into a small viewport.

To change the default setting, set the *AdjustWidgetSize* variable in the *glg\_hmi\_config* configuration file.

If an application uses custom widgets of a defined size which are not supposed to be resized by the user, the *Adjust Widget Size* feature may be disabled to make sure that all widgets appear in their original design size regardless of the size of the GLG Editor window. If widget size adjustment is enabled, the widgets inserted into the drawing when the editor window is maximized will appear in a different size compared with the widgets that were inserted when the editor window was not maximized.

### ***Make Widget Viewports Transparent***

The *Make Widget Viewports Transparent* option may be used to add dials and gauges with transparent background. If it is checked, adding a viewport-based widget from the palette converts it to use a light viewport with a transparent background. If the option is not checked, a viewport object with non-transparent background is used.

### ***Use Light Viewports***

The *Use Light Viewports* option may be checked to convert viewport-based widgets to use light viewports when the widgets are added to the drawing.

## ***Custom Objects***

The *Custom Objects* palette displays samples of pre-built widgets that can be used in a drawing. It contains buttons, sliders and other objects.

Refer to the *Using Custom Object Palette* section on page 70 of the *GLG Builder and Animation Tutorial* for detailed description of all widgets in the *Custom Objects* palette.

## ***Custom Widget Samples***

The *Custom Widget Samples* palette displays samples of HMI components that use application-specific public properties. Public properties provide a simplified interface for configuring components via the *Properties* dialog.

Click on a component in the palette to add it to the drawing, then use *Object, Properties* to display its public properties.

## ***Read Palette***

The *Read Palette* option can be used to load a palette into the HMI Configurator. A palette is defined by a *Palette Description File* with the *.pal* extension. This file provides information about the palette and the objects it contains.

To load a palette, select *Read Palette*, then select the palette's *.pal* file using the activated file browser.

## ***Read Directory***

The *Read Directory* option can be used to scan a directory containing GLG drawings and display a palette containing all drawings in the directory. To read a directory, select *Read Directory*, then select a directory to read with the activated file browser.

## ***Edit***

The *Edit* Menu provides options that let you make and manipulate copies of objects.

These options operate on the selected object. To operate on several objects at the same time, use the menu options for selecting multiple objects, then use editing options.

The *Cut*, *Copy*, and *Paste* options add and remove objects. The *Clone* options let you position and transform the added objects as they are created.

## ***Select Multiple Objects***

*Select Multiple Objects* is equivalent to using *Ctrl*-click. It starts multiple object selection without the need to hold the *Ctrl* key. After the option is selected, click on the objects in the drawing to add or delete them from the selection.

### ***Select Rectangular Area***

*Select Rectangular Area* is equivalent to clicking and dragging the mouse in the drawing to define the selection rectangle. It provides a convenient option for starting a rectangular selection for cases when all drawing area is covered with objects and there is no free space to click and drag the mouse without selecting some object. After the option is selected, click and drag the mouse anywhere in the drawing area to define the selection rectangle. All objects that are either completely or partially enclosed by the rectangle will be selected.

### ***Select All***

*Select All* selects all objects in the drawing.

### ***Cut***

*Cut* removes the selected object from the drawing and places it on the clipboard. The cut object remains on the clipboard until you replace it by performing another *Edit*, *Cut* or an *Edit*, *Copy* operation.

You can retrieve the cut object by using *Edit*, *Paste*.

### ***Copy***

*Copy* places a copy of the selected object on the clipboard without removing it from the drawing. The copied object remains on the clipboard until you replace it by performing an *Edit*, *Cut* or *Edit*, *Copy* operation.

You can retrieve the copied object by using *Edit*, *Paste*.

### ***Paste***

*Paste* gets a cut or copied object from the clipboard and adds it to the current drawing. Pasting an object does not delete the object from the clipboard, so you can paste the same object repeatedly.

### ***Delete***

*Delete* removes the selected object from the drawing. The deleted object is irrevocably discarded; it is *not* placed on the clipboard.

To remove an object from the drawing and place it on the clipboard, use *Edit*, *Cut*. This allows you to move an object from one part of the drawing to another.

### ***Define Clone Offset***

*Define Clone Offset* determines the placement of a copy with respect to the original object.

When you select *Define Clone Offset*, the HMI Configurator prompts you to click on two points that define the clone path. The clone path specifies the distance and direction from the origin of the original object to the origin of the copy.

The HMI Configurator uses a default offset of 50 units to the Southeast for all objects unless you redefine the offset.

To copy an object using the defined offset, use the *Copy With an Offset* option of the *Edit* menu.

### ***Copy Object With an Offset***

*Copy Object With an Offset* creates a copy of the selected object. The copy is positioned with an offset defined by the *Define Clone Offset* option.

### ***Edit Group Elements***

Provides options for editing multiple objects in a group via resources. A selected editing operation, such as *Set D Resource*, can be applied to elements of the group, skipping the elements that do not have the specified resource. If the *Recursive* option is selected, the editing operation will be also applied to the sub-elements inside each group element. The option may be used to set resources of either a group or a single object.

#### *Set D Resource*

Traverses all objects in the group and sets the value of a specified resource of D (double) type.

#### *Set G Resource*

Traverses all objects in the group and sets the value of a specified resource of G type (XYZ coordinate or color RGB).

#### *Set S Resource*

Traverses all objects in the group and sets the value of a specified resource of S (String) type.

#### *Recursive*

Controls traversing of subobjects inside group elements that are containers, such as groups, viewports and light viewports. If the *Recursive* option is set, the editing operation will be applied to the elements of the group, as well as sub-elements inside each container element with an unset *HasResources* flag. If the option is unset, the editing operation will be applied only to the top-level elements of the group.

### ***Reset Scaling Xform***

For a reference object, *Reset Scaling Xform* resets the internal transformation the reference object uses to stretch its instance, when the bounding box is stretched with the mouse. Selecting *Reset Scaling Xform* restores the original, unstretched appearance of the instance.

### ***Run-Time Scrollbars***

The *Run-Time Scrollbars* menu has options for setting the drawing's run-time scrollbars. The run-time scrollbars are displayed when the drawing is deployed in an application and used to scroll the drawing.

The menu has the following options:

#### *None*

Disables the drawing's run-time scrollbars.

#### *X and Y*

Enables the drawing's run-time scrollbars, both in the X and Y direction.

#### *X*

Enables the drawing's horizontal run-time scrollbar.

#### *Y*

Enables the drawing's vertical run-time scrollbar.

### ***Keep Edit Ratio***

Provides access to the *Keep Edit Ratio* attribute of the top drawing viewport. If checked, the drawing will maintain its X/Y ratio to prevent shape distortion of circles and other objects.

## ***Undo***

### ***Undo***

*Undo* reverts the effect of the last editing operation, such as changing geometry or an attribute value of an object or group of objects, changes to object's control points, layout and alignment operations, zooming and panning and others. The undo button displays the last editing operation that can be undone and changes its label to *Redo* after performing the *Undo* operation. For some editing operations, such as deleting objects, the *Paste* option should be used instead of *Undo*.

### ***Undo History***

*Undo History* displays a list of recent editing actions and allows selecting individual actions to undo or redo.

Some geometry editing actions require the drawing's viewing state (pan and zoom) to be unchanged in order to be reverted properly. The changes to the drawing's viewing state are listed in the *Undo History* list and can be undone as well.

## **View**

The View Menu provides options to zoom and pan the drawing.

### ***Zooming***

The Zooming submenu provides options to let you change the scale of the view. The changes in scale affect your view of the drawing, not the drawing itself, and are saved with the viewport. The HMI Configurator's *Control Panel* also provides zooming controls.

#### ***Zoom In***

*Zoom In* increases the scale of the drawing, so the objects look bigger. Zooming into a drawing enlarges the objects in the center of the drawing, but objects near the edges of the drawing may be clipped.

To control the degree of scaling, use *View, Zooming, Set Zoom Factor*.

#### ***Zoom Out***

*Zoom Out* decreases the scale of the drawing, so the objects look smaller.

To control the degree of scaling, use *View, Zooming, Set Zoom Factor*.

#### ***Zoom To***

*Zoom To* zooms into a specified area of the drawing. When you select this option, the HMI Configurator prompts you to specify two points that define a bounding box for the zoom area.

To recover your view of the excluded area, use *View, Zooming, Zoom Out*.

#### ***Set Zoom Factor***

*Set Zoom Factor* controls the scale factor to be applied when you zoom using the *Zoom In* and *Zoom Out* buttons of the *Control Panel* or the menu options. The HMI Configurator prompts you for a scaling factor (for example, 2 to scale up or 0.5 to scale down).

#### ***Reset Zoom***

*Reset Zoom* returns to the normal scale for the view (100% zoom).

### ***Pan To***

*Pan To* moves the center of the view to another part of the drawing area.

After you select this option, click on a spot to use as the new center of the view.

### ***Scroll by Dragging***

Starts dragging mode. Click and drag the mouse after selecting this option to scroll the drawing with the mouse.

Scrolling the drawing with the mouse may also be performed by the *Ctrl-click-drag* sequence in any empty area of the drawing. However, if the drawing area is completely occupied by objects, this menu option provides an alternative.

### ***Traverse***

The *Traverse* Menu provides options for editing inner objects of a widget. For example, a real-time chart widget is comprised of a Chart object inside the widget's viewport. The Resource Browser may be used to edit the chart's properties as resources. To edit the chart's properties via the Properties dialog, the *Set Focus* menu may be used to set editing focus inside the widget's viewport, after which the chart object can be selected for editing.

### ***Set Focus***

*Set Focus* sets editing focus into the viewport selected with the mouse. To set the focus, use this menu option (or click on the *Set Focus* button on the *Control Panel*). The HMI Configurator prompts you to select a viewport to set focus into. *Ctrl-Shift*-clicking on a viewport also moves focus into it, acting as a convenient shortcut for *Set Focus*. If a viewport is part of a group, the first *Ctrl-Shift*-click on a viewport selects it, and the second *Ctrl-Shift*-click moves focus inside the viewport.

To return to the default editing mode, use *Traverse*, *Main Focus*, or click on the *Main Focus* button.

### ***Main Focus***

*Main Focus* returns focus to the main drawing area. *Ctrl-Shift*-clicking on a top-level viewport of the drawing area also moves focus into it.

This option is grayed out until you use *Traverse*, *Set Focus* to change the viewport editing mode.

### ***Arrange***

The *Arrange* Menu provides options to let you change the relationships between objects.

### ***Create Temporary Group***

*Create Temporary Group* is equivalent to *Select Rectangular Area* described on page 60.

### ***Select Multiple Objects***

*Select Multiple Objects* is equivalent to *Select Multiple Objects* described on page 59.

### ***Add Object to Group***

*Add Object to Group* adds an object to the selected group object. It may be used with both temporary and permanent groups.

Use the following procedure:

- Select the group or create a temporary group by *Ctrl-clicking* on objects. in the drawing.
- Select *Arrange, Add Object to Group*.
- Click on the first object to add to the group.
- Click on the next object to add to the group.
- When you have added all the objects to the group, use the *Esc* key or the right mouse button to complete the operation.

The status bar at the bottom of the HMI Configurator window provides prompts to guide you through the procedure.

This option is equivalent to the *Add Object* button in the *Properties* dialog for a group object.

### ***Delete Object from Group***

*Delete Object from Group* removes a single object from the selected group. The option may be used with both temporary and permanent groups.

Use the following procedure:

- Select the group.
- Select *Arrange, Delete Object from Group*.
- Click on the first object to remove from the group.
- Click on the next object to remove from the group.
- When you have removed the objects from the group, use the *Escape* key or the right mouse button to complete the operation.

The status bar at the bottom of the HMI Configurator window provides prompts to guide you through the procedure.

This option is equivalent to the *Remove Object* button in the *Properties* dialog for a group object.

### ***Add or Delete Object from Group***

*Add or Delete Object from Group* adds an object to the group if the object is not part of the group, or deletes the object from the group if the object is a part of the group. The option may be used with both temporary and permanent groups.

Use the following procedure:

- Select the group.

- Select *Arrange, Add or Delete Object from Group*.
- Click on the first object to add or delete from the group.
- Click on the next object to add or delete from the group.
- When you have removed the objects from the group, use the *Esc* key or the right mouse button to complete the operation.

The status bar at the bottom of the HMI Configurator window provides prompts to guide you through the procedure.

For temporary groups, the same action can be accomplished by *Ctrl*-clicking on the object with the mouse.

### ***Select Next***

Select Next is used to select individual objects inside the group for editing. When you click on an object in a group with the mouse, you select the entire group. If you want to edit only one member of the group, you can use *Select Next* to select it.

When a permanent group is selected, the *Ctrl-Shift*-click on an object in the group selects the object, acting as a shortcut for *Select Next*. When an object inside the group is selected using *Select Next*, the boundaries of its parent group are highlighted with a dotted line to provide visual feedback for the group.

This option is equivalent to the *Select Next* button in the *Properties* dialog for a group object.

*Select Next* mode is aborted when an object outside of the group is selected.

### ***Edit All (First)***

For a group object, *Edit All (First)* starts editing the attributes of objects in the group by using the first object in the group to select a set of attributes for editing. This is a convenient option for fast editing of groups that contain objects of the same type.

For groups that contain objects of different types, the *Edit All (Select)* option allows you to select a set of attributes to edit. For example, if the group contains both the polygon and text objects, the *Edit All (Select)* option allows you to select the polygon or text attributes to be edited.

This option is equivalent to the *Edit All (First)* button in the *Properties* dialog for a group object.

### ***Edit All (Select)***

For a group object, *Edit All (Select)* allows you to choose a set of attributes to edit. This may be convenient when the group contains objects of different types with different sets of attributes.

When you select this option, the HMI Configurator prompts you to select an object that has the attributes you want to change. The changes you make to an attribute in the *Properties* dialog apply to all the objects in the group that have the attribute.

For example, consider a group that contains two circles, a parallelogram, and a fixed text object. Select *Edit All (Select)* and then select a circle. The *Properties* dialog shows the attributes for a circle object. Resetting the *Resolution* affects both circles. Resetting the *LineWidth* affects both of the circles and the parallelogram. However, the text object doesn't have a *LineWidth* attribute and is not affected.

This option is equivalent to the *Edit All (Select)* button in the *Properties* dialog for a group object.

### ***Permanent Group***

*Permanent Group* toggles the type of the selected group object between temporary and permanent. The current group type is displayed as the state of this toggle button.

### ***Explode Group***

*Explode Group* discards the permanent group, so that the objects in a group become independent objects. Any dynamics attached to the group is discarded with the group.

### ***Reorder***

The *Reorder* submenu provides options to let you change the drawing order of objects.

The last object drawn appears to be in front of any other overlapping objects, with the exception of components that use viewports: viewports are windows and always appear on top of graphical objects.

#### *Move to Back*

*Move to Back* moves the selected object behind other objects in the drawing. All overlapping objects appear to be in front of the selected object.

#### *Bring to Front*

*Bring to Front* moves the selected object in front of other objects in the drawing. All overlapping objects appear to be behind the selected object.

#### *Move Backward*

*Move Backward* changes the selected object's place in the rendering list, moving the object back in the list by one position every time *Move Backward* is selected (the *Ctrl+* accelerator may also be used).

#### *Move Forward*

*Move Forward* changes the selected object's place in the rendering list, moving the object forward in the list by one position every time *Move Forward* is selected (the *Ctrl-* accelerator may also be used).

## **Convert Viewport**

Contains options for converting viewports.

### *Viewport -> Light Viewport*

Converts a selected viewport to a light viewport.

### *Light Viewport -> Viewport*

Converts a selected light viewport to a viewport.

## **UTF8 Conversions**

### *Convert Strings from Current Local to UTF8*

Converts string attributes (such as *TextString* of a text object) of all objects in the drawing from the current locale to UTF8 and sets the *UTF8Encoding* flag of each attribute to YES.

### *Convert UTF8 Strings to Current Locale*

Converts string attributes (such as *TextString* of a text object) of all objects in the drawing from UTF8 encoding the current locale and sets the *UTF8Encoding* flag of each attribute to NO.

## **Layout**

The *Layout* Menu provides options to align and layout objects in the drawing. It may also be used to view or set the objects' width and height using the *Layout Toolbox* option.

### **Layout Toolbox**

This option activates the *Layout Toolbox* which contains icons and controls for performing various align and layout operations. See the *Object Layout and Alignment* chapter on page 47 for more information.

### **Position in Drawing**

This submenu provides options for positioning object in the drawing's viewport. The *Align To Viewport Span* option of the *Layout* menu controls the area used for alignment: the viewport span or the currently visible area of the viewport.

#### *At Left Edge*

*At Left Edge* aligns the left edge of the selected object with the left edge of the drawing.

#### *Center Horizontally*

*Center Horizontally* positions the selected object in the center of the drawing in the horizontal direction.

### *At Right Edge*

*At Right Edge* aligns the right edge of the selected object with the right edge of the drawing.

### *At Top Edge*

*At Top Edge* aligns the top edge of the selected object with the top edge of the drawing.

### *Center Vertically*

*Center Vertically* positions the selected object in the center of the drawing in the vertical direction.

### *At Bottom Edge*

*At Bottom Edge* aligns the bottom edge of the selected object with the bottom edge of the drawing.

## ***Set Width/Height***

The *Set Width/Height* submenu provides options for setting an object's width and height.

### *World Coordinates*

Provides options for setting an object's width and height in world coordinates.

#### *Set Width*

#### *Set Height*

Opens the Layout Toolbox dialog and selects options for setting width or height of the selected object in world coordinates. The current width or height is displayed in the dialog and can be changed by entering a new value.

#### *Set Group Element Width*

#### *Set Group Element Height*

Opens the Layout Toolbox dialog and selects options for setting width or height of all elements of the selected group in world coordinates. The dialog displays width or height of the first element of the group. The width or height of all group elements may be changed by entering a new value.

### *Pixels*

Provides options for setting an object's width and height in pixels.

#### *Set Width*

#### *Set Height*

#### *Set Group Element Width*

#### *Set Group Element Height*

These options are the same as the corresponding options for setting width/height in world coordinates, except that they use pixels instead of the world coordinates.

## ***Set Space/Distance***

The *Set Space/Distance* submenu provides options for setting space or distance between elements of a group. The distance is measured between the object's centers, while space is the distance between the object's extents.

The *Align Points* option controls how object extents are determined, using either an object's bounding box or control points.

### *World Coordinates*

Provides options for setting space or distance between elements of a group in world coordinates.

*Set Horiz. Space*

*Set Vert. Space*

*Set Horiz. Center Distance*

*Set Vert. Center Distance*

Opens the Layout Toolbox dialog and selects options for setting space or distance between elements of the selected group in world coordinates. The space or distance can be changed by entering a new value instead of the default value displayed in the dialog.

### *Pixels*

Provides options for setting space or distance between objects in a group in pixels.

*Set Horiz. Space*

*Set Vert. Space*

*Set Horiz. Center Distance*

*Set Vert. Center Distance*

These options are the same as the corresponding options for setting space/distance in world coordinates, except that they use pixels instead of the world coordinates.

## ***Align***

The *Align* submenu provides options for aligning objects.

### *Align Left*

*Align Left* aligns the left edge of each selected object with the left edge of the anchor object. If no anchor object is selected, the left most selected object is used as an anchor.

### *Align Horiz. Center*

*Align Horiz. Center* aligns the center of each selected object with the center of the anchor object horizontally. If no anchor object is selected, the left most selected object is used as an anchor.

### *Align Right*

*Align Right* aligns the right edge of each selected object with the right edge of the anchor object. If no anchor object is selected, the right most selected object is used as an anchor.

### *Align Top*

*Align Top* aligns the top edge of each selected object with the top edge of the anchor object. If no anchor object is selected, the highest selected object is used as an anchor.

### *Align Vert. Center*

*Align Vert. Center* aligns the center of each selected objects with the center of the anchor object vertically. If no anchor object is selected, the highest selected object is used as an anchor.

### *Align Bottom*

*Align Bottom* aligns the bottom of each selected object with the bottom of the anchor object. If no anchor object is selected, the lowest selected object is used as an anchor.

## ***Make Same Size***

*Make Same Size* submenu provides options for setting size of selected objects to be the same.

### *Width*

Sets the width of each selected object to be the same as the width of the anchor object. If no anchor object is selected, the left most selected object is used as an anchor.

### *Height*

Sets the height of each selected object to be the same as the height of the anchor object. If no anchor object is selected, the left most selected object is used as an anchor.

### *Both*

Sets both the width and height of each selected object to be the same as the width and height of the anchor object. If no anchor object is selected, the left most selected object is used as an anchor.

## ***Distribute***

*Distribute* submenu provides options for distributing objects in the selected group, leaving no spaces between the objects.

### *Across*

Distributes objects horizontally leaving no extra spaces between the object's extents.

### *Down*

Distributes objects vertically leaving no extra spaces between the object's extents.

## ***Space Evenly***

*Space Evenly* submenu provides options for evenly distributing spaces between objects in the selected group.

*Across*

Distributes objects horizontally with equal spaces between the object's extents.

*Down*

Distributes objects vertically with equal spaces between the object's extents.

***Distribute Evenly***

*Distribute Evenly* submenu provides options for evenly distributing objects in the selected group using objects' centers.

*Across*

Distributes objects horizontally with equal distance between the objects' centers.

*Down*

Distributes objects vertically with equal distance between the objects' centers.

***Select Anchor***

*Select Anchor* defines the anchor object. Select this option, then click on the object with the mouse to define it to be the anchor. The anchor selection will be preserved until the currently selected group is unselected.

***Align Points***

When this toggle is checked, the objects' control points are used to align objects. If the toggle is unchecked, the objects' extents will be used.

***Align To Viewport Span***

When this toggle is checked, the viewport span is used as the drawing extent used to position the object. If the toggle is unchecked, the currently visible area of the viewport's window is used as the drawing extent.

For resizable viewports, the span is defined by the *SpanX* and *SpanY* attributes of the viewport's screen object. For fixed scale viewports, the span is the size of the viewport. By default, the span extent is annotated with a different color and span marks at the corners of the span.

For resizable viewports with stretch, as well as fixed scale viewports, the difference between the span and the current visible area is noticeable only when the viewport is zoomed or edited by using *Hierarchy Down*.

## *More*

This option is equivalent to the *Layout Toolbox* option: it brings the Layout Toolbox for more alignment and layout options.

## **Object**

The *Object* Menu provides options to let you create and manipulate objects within the drawing.

### *Create*

The options under the *Create* option let you add new objects to the drawing.

Most of the options under the *Create* submenu correspond to the buttons in the *Object Palette* on the left side of the HMI Configurator window. You can create the object by choosing its type from this menu, or by clicking on the corresponding button in the palette.

If an object has an icon in the drawing primitives palette, the icon is shown next to the object description. If an icon is not displayed, the object may be created only by using the *Object, Create* main menu.

See the *Creating Objects* chapter for instructions on drawing objects.

### *Polygon*

The *Polygon* options let you create the following kinds of polygons:

- Open polygon 
- Closed polygon 
- Filled polygon 

To create any type of polygon, click in the drawing to define the polygon's points, then click the right mouse button (or use the *Escape* key) to complete the polygon. For the closed and filled polygons, the HMI Configurator joins the first and last vertices, closing the polygon.

### *Rectangle*

The *Rectangle* options let you create the following kinds of rectangles:

- Rectangle 
- Filled rectangle 

Although the GLG object set does not include a rectangle object, the *Rectangle* options are provided for convenience. Drawing a rectangle actually creates a parallelogram with perpendicular sides.

To create a rectangle, click on two points to define the diagonal corners of the rectangle.

### Parallelogram

The *Parallelogram* options let you create the following kinds of parallelograms:

- Parallelogram
- Filled parallelogram

To create a parallelogram, click on one point, and then click on two other points. The second and third points define two vectors from the first point; they specify the angles and lengths of the opposing sides.

### Rounded Rectangle

The *Rounded Rectangle* options let you create the following kinds of rectangles:

- Rounded Rectangle
- Filled rounded rectangle 

To create a rounded rectangle, click on two points to define the diagonal corners of the rectangle. It creates a parallelogram with perpendicular sides and rounded corners. The size of the rounded corners is controlled by the object's *Radius1* and *Radius2* attributes.

### Arc

The *Arc* options let you create the following kinds of arcs:

- Arc 
- Chord arc 
- Segment arc 
- Filled chord arc 
- Filled segment arc 

An arc is a many-sided regular polygon, like a circle; however, an arc does not encompass 360°. A simple arc is just the segment of a circle's perimeter connecting the points you choose. A chord arc closes the shape with a straight line from one end of the arc to the other. A segment arc closes the shape with two straight lines from each end of the arc to the center, like a wedge of pie.

To create any type of arc, click on the center point, a point to define the radius, and a point to define the angle of the sector.

You can convert an arc to a circle by editing its *StartAngle* or *EndAngle* attributes.

The control points of an arc or a circle are unusual for graphical objects. See the description of a circle (below) for an explanation.

## Circle

The *Circle* options let you create the following kinds of circles:

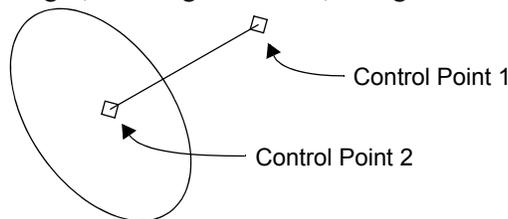
- Circle 
- Filled circle 

A circle is a many-sided regular polygon. Its *Resolution* attribute specifies the number of sides used to render the circle. At low *Resolution* values, the shape no longer resembles a circle; for example, a circle with a *Resolution* of 5 is a pentagon.

To define a circle, click on the center point, then on a point to define the radius.

In GLG, both circles and arcs are rendered using an *Arc* object type. A circle is a special case of an arc with a *StartAngle* of 0 and an *EndAngle* of 360.

Circles and arcs have an arrangement of control points different from other graphical objects. Each circle has two control points that initially appear superimposed at the center of the circle. (The move point is moved slightly away from the center to avoid confusion.) The two points define a line to which the circle is perpendicular. The length of the line is ignored. This means that you can grab one of the control points, and tilt the circle by moving it. However, a circle has no control points on its perimeter for resizing it; to change its radius, change its *Radius* attribute or use the *Resize Box*.



**The Control Points of a Circle**

## Ellipse

The *Ellipse* options let you create the following kinds of ellipses:

- Ellipse
- Filled ellipse 

An ellipse is a rounded rectangle with rounded corners taking the whole extent of the rectangle. Its *Resolution* attribute specifies the number of segments used to render each corner of the ellipse. At low *Resolution* values, the shape no longer resembles an ellipse; for example, an ellipse with a *Resolution* of 2 is a hexagon.

To create an ellipse, click on two points to define the diagonal corners of the ellipse's bounding rectangle.

### *Spline*

The *Spline* options let you create a multi-point spline used to render curves in 2D or 3D space:

- Spline 
- Filled spline 

To create a spline, click in the drawing area to define the number of spline control points, then click the middle or right mouse buttons (or use the *Escape* key) to complete the spline. The spline will render a smooth curve defined by the control points. The shape of the curve may be changed by moving the control points.

### *Marker*

A marker is an object that indicates the position of a single point. Markers are made by selecting one or more shapes from a set of predefined shapes such as squares, crosses, and circles. Unlike other graphical objects, marker objects do not change their size when the viewport is resized. 

To create a marker, select *Object, Create, Marker* and then specify the point.

Since markers are always drawn the same size, they are not affected by transformations. To change a marker's size, change its *Size* attribute.

### *Image*

An image object may be used to display an image in the GIF or BMP (on Windows only) formats. The *Image* options let you create the following kinds of images:

- Fixed Size Image 
- Scalable Image 

To create an image, select the type of the image: fixed size or scalable, define its position (one control point for the fixed image and two points for the scalable image), and select the image file.

For images of fixed size, the *Anchoring* attribute may be used to control the image's position relative to its control point.

### *Text*

The *Text* options let you create the following kinds of text objects:

- Fixed Text presents a string. To create a fixed text object, click in the drawing to define the text's position, then type the string in the text entry box. 

- Scaled Text presents a string within a bounding box. Resizing the bounding box changes the font size of the string. To create a scaled text object, click on two points to define the diagonal corners of the bounding box, then type the string in the text entry box. Note that the text is scaled by selecting different size fonts from the viewport's font table, not by changing the dimensions of the characters. Depending on the selection of sizes defined in the font table, this may appear to give incorrect results when the text object is resized. 
- Spaced Text presents a string within a bounding box, with flexible orientation. Changing the box repositions the string. To create a spaced text object, click on two points to define a line, then click on a third point to define the height of the bounding box. Finally, type the string in the text entry box. . The third control point is used for line positioning when the text object has several lines.

A text object is a graphical object that presents a string. The text object itself just sets the boundaries of the text. The text object's data attribute defines the string that appears in the drawing.

### GIS Object

A GIS Object provides a way to embed GIS maps generated with the GLG Map Server into GLG drawings. It automatically handles all aspects of the low-level interaction with the Map Server to display, zoom and pan GIS map data. The GIS Object  provides attributes to control projection, center and extent of the map.

To create a GIS Object:

1. Select *Object, Create, GIS Object* and choose rectangular or orthographic projection to create the GIS Object.
2. Click on two points to define a rectangular area to use for the map display.
3. When prompted, select a dataset file that describes the GIS data to render.

The GIS Object will display the map specified by the dataset file. The map may be positioned by changing the *GISCenter* and *GISExtent* attributes of the object. The GIS Zoom mode of the drawing's Integrated Zoom feature may be used at run time to zoom and pan the map.

### Connector

This option creates a connector object which may be used to connect other objects in the drawing. It is useful when connecting objects in a diagram drawing.

There are two types of connectors:

- A Recta-Linear connector connects objects with linear segments, maintaining right angles between adjacent segments. There are Horizontal  and Vertical  recta-linear connectors.
- An arc connector connects objects with an arc .

To create a connector, select the connector type, then click in the drawing to define its shape. For the arc connector, select 3 points to define the arc. For the recta-linear connector, select any number of points to define one or more recta-linear segments and press the *Escape* key or the middle mouse button to finish.

### ***Edit Toolbox***

This option activates the *Edit Toolbox* for fast access to the selected objects' properties. Refer to the *Editing Objects with the Edit Toolbox* section on page 20 for more information.

### ***Properties***

*Properties* displays a dialog that lists the attributes of the selected object and provides access to their dynamics, alarms and tags.

### ***Public Properties***

*Public Properties* display a dialog that lists user-defined public properties of a component.

### ***Resources***

*Resources* displays a *Resource Browser* that shows resources of the selected object. If no object is selected, resources of the whole drawing (at the current level of the hierarchy) are shown. Selecting a different object updates the Resource Browser to display the object's resources.

### ***Tags***

*Tags* brings a *Tag Browser* that shows a list of tags of the selected object, or of the whole drawing if no object is selected. Selecting a different object updates the *Tag Browser* to display the object's tags.

### ***Alarms***

*Alarms* brings an *AlarmBrowser* that shows a list of alarms attached to the selected object or all alarms of the whole drawing if no object is selected. Selecting a different object updates the *Alarm Browser* to display the object's alarms.

## ***Object Dynamics***

The *Object Dynamics* menu contains options for adding and editing geometrical dynamics (such as move, scale or rotate) attached to the object.

### *Add Dynamics*

*Add Dynamics* adds geometrical dynamics (such as move, scale or rotate) to the selected object. Selecting the option displays a menu of available dynamics types.

Refer to the *Adding Geometrical Dynamics* chapter on page 32 for details.

### *Edit Dynamics*

*Edit Dynamics* brings the dialog for editing geometrical dynamics attached to the object.

## ***Tooltip***

The *Tooltip* menu contains options for adding and editing object tooltips.

### *Add Tooltip*

*Add Tooltip* attaches a tooltip action to the selected object and opens the action's Properties dialog for entering the tooltip string as a value of the action's *Tooltip* attribute.

To activate tooltip processing, the *ProcessMouse* attribute of the viewport containing the object has to include the *Tooltip* mask. The *Run* mode of the Builder may be used to test the tooltips.

### *Edit Tooltip*

*Edit Tooltip* opens the Properties dialog for editing the tooltip action attached to the selected object.

### *Delete Tooltip*

*Delete Tooltip* deletes a tooltip action attached to the selected object.

## ***Actions***

The *Actions* submenu provides options for adding and editing actions and commands attached to objects in the drawing.

### *Add Command*

*Add Command* attaches a command action to the selected object. It prompts the user to select a type of command from a list of available commands, then opens a dialog for specifying values of the command's data properties.

The *Back* button may be used to return to the action's Properties dialog after the command data has been entered. The action properties specify the event that triggers the command's execution. Refer to the *Action Object* chapter on page 219 for more information.

### *Add Input Command*

*Add Input Command* attaches an input command action to the selected input object. It uses the same user interface as the *Add Command* option described above but is active only when an input object, such as a button, a toggle or a slider, is selected.

### *Edit Actions*

*Edit Actions* opens an Action List dialog for editing actions attached to the selected object. The dialog provides an interface for editing attributes, reordering or deleting individual actions. Refer to the *Action Object* chapter on page 219 for more information.

The *Actions* button in the *Status Panel* at the bottom of the drawing area may be used as a shortcut.

### *Delete All Actions*

*Delete All Actions* deletes all actions attached to the selected object. To delete individual actions, use the *Edit Actions* option, select an action to be deleted, then press the *Delete* button.

### *Mark Actions*

*Mark Actions* stores a copy of the selected object's actions in the clipboard to subsequently add them to other objects in the drawing. To select individual actions instead of the list of all actions attached to the object, use the *Copy* button in the action's Properties dialog.

### *Paste Actions*

*Add Marked Actions* adds a copy of the previously marked actions to the selected object.

## ***Disable User Interaction***

*Disable User Interaction* removes an interaction handler from the selected widget. It can be used to disable user interaction for widgets such as dials or meters, making them output-only.

## **Run**

The *Run* menu provides options to let you test the animation of drawings.

### ***Start***

*Start* starts the Run mode.

*Start* also displays a dialog for entering an animation command.

Refer to the the *Prototyping Dynamic Behavior Using Run Mode* chapter on page 44 for information on animation commands and using live data to animate.

In the Run mode, the Builder activates the *Run Toolbar* that contains the following controls:

- **Updates** - displays the number of updates performed from the start of the Run mode.

- **Seconds** - shows the number of seconds the Run mode was active; this does not include the time when the Run mode was paused.
- **Updates/sec** - displays an average number of updates per seconds. Clicking on this control changes it to **Secs/Update**, which displays an average number of seconds it takes to execute one update.
- **Pause / Resume button** - temporarily pauses drawing updates without quitting the Run mode. Pressing the button the second time resumes updates again.
- **Stop button** - stops the Run mode.
- **Update Speed** - controls the frequency of updates, from 9 (maximum) to 1 (minimum).

## *Stop*

*Stop* halts the Run mode.

## *Options*

The *Options* Menu provides options to let you customize the operation of the HMI Configurator. The *glg\_hmi\_config* configuration file can be used to specify default values of various options.

## *Draw Grid*

The *Draw Grid* submenu provides options to let you display grid lines with a drawing. The grid is displayed at a selected grid interval. In addition to the constant grid spacing options, the *Adaptive (Constant Space)* option automatically selects a grid interval to maintain approximately constant visual grid spacing. To disable grid, use the *No Grid* option.

The grid is not saved as part of the drawing. It is just a convenience for editing a drawing.

To position objects relative to the grid, use *Options, Snap To, Grid*.

## *Snap To*

The *Snap To* submenu provides options to let you align objects. Snapping places objects by rounding their coordinates as you draw them.

The options on the *Snap To* submenu specify the resolution of the alignment grid. If you specify *Snap to Grid*, the HMI Configurator snaps to the grid and also to the midpoints of the grid. The *0 (No snapping)* option is used to disable snapping, and *1 (Round to integer)* option is used to get rid of the fractional coordinate values, which is almost the same as no snapping but with integer coordinates.

## *Show Axis*

*Show Axis* shows or hides the axis icon. The axis icon indicates the center and orientation of the current view and coordinate system.

To change the view projection and the coordinate system used to view the drawing, use the options on the *View* menu.

### ***Show Coordinates***

*Show Coordinates* shows or hides the mouse coordinate display window in the lower right corner of the HMI Configurator. Both the screen (S) and world (W) coordinates are displayed.

### ***Show Span***

#### *Span Marks*

Shows or hides red markers that display the extent of the drawing area in the unzoomed state.

#### *Span Area*

Enables or disables a color annotation of the extent of the drawing area in the unzoomed state.

### ***Save Format***

*Save Format* determines the format the HMI Configurator uses to save a drawing to a file. The HMI Configurator can save files in two formats:

- Binary, which loads quickly but is not portable across platforms that use different binary data representations.
- ASCII, which is completely portable across platforms but loads slightly slower than binary.

The default format for the saved drawing is ASCII. This is the recommended format that works not only across all hardware platforms but also across all GLG deployment options: C/C++, Java, C# and JavaScript.

### ***Save Compressed***

*Save Compressed* enables or disables saved drawing compression. When the drawing compression is enabled, drawings are saved using gzip-compression. The saved file extension is not changed when the drawing is saved compressed. This option controls only saving operations: on loading, the drawing format is recognized and handled automatically.

The default is *Save Compressed*.

### ***SaveImageFormat***

*SaveImage Format* specifies the image format to use when saving images via the *Save Image* menu option. Supported formats are JPEG and PNG.

### ***Convert Strings to UTF8 on Save***

When this option is checked, all string attributes (such as *TextString* of a text object) are automatically converted from the current locale to UTF8 when the drawing is saved. This is especially convenient when using localized strings on Windows in a drawing that will be used on the web with the JavaScript. JavaScript supports a limited number of character sets, using UTF8 as a default character set, while the UTF8 system locale is not supported on Windows. The *Convert Strings to UTF8 on Save* option provides a convenient way to save the drawing in the UTF8 locale without a need to use external converters.

The *Arrange, UTF8 Conversions* menu provides options for converting all strings in the drawing from the current locale to UTF8 and from UTF8 to the current locale. These options may be used to verify the converted drawing before saving it.

When a string attribute is converted to UTF8, its *UTF8Encoding* flag is set to YES.

### ***Selection Options***

#### *Show Attachment Points*

Enables or disables display of attachment points of the selected reference object.

#### *Disable Dynamics For Editing*

*Disable Dynamics For Editing* disables geometrical dynamics (such as move, scale or rotate) when the object is selected. When the option is set, the object will appear in its untransformed state with dynamics disabled. This is convenient for editing objects in the object's initial state.

The option disables only the geometrical dynamics attached to the object. It does not disable dynamics attached to the object's control points.

#### *Select Non-Components*

*Select Non-Components* modifies the behavior of the *Select All* option: if *Select Non-Components* is set, *Select All* selects only objects that do not have public properties defined (non-component objects).

### ***Trace / Highlight***

#### *Trace Tags*

When activated, *Trace Tags* highlights all objects in the drawing that depend on a tag selected in a Tag Browser.

#### *Trace Resources & Constraints*

When activated, *Trace Resources & Constraints* highlights all objects in the drawing that depend on a resource selected in a Resource Browser or an attribute displayed in the Attribute Object dialog.

### *Highlight Widgets*

Controls tracing behavior. If *Highlight Widgets* is unchecked, the first found drawable parent of the traced object will be highlighted. If it is checked, the parent that has the *IsWidget* resource will be highlighted.

### *Highlight Top Level Objects*

Controls tracing behavior. If *Highlight Top Level Objects* is unchecked, the first found drawable parent of the traced object will be highlighted. If it is checked, the parent that is a direct child of the Drawing Area that has editing focus will be highlighted.

## ***Magnetic Control Points***

Controls the Magnetic Mode.

### *Attach to Point*

Attaches a new point to the highlighted point selected with the mouse click.

### *Use Point Position*

Positions a new point at the position of the highlighted point selected with the mouse click.

### *Disable*

Disables Magnetic Mode. A new point will be positioned at the exact mouse click position without attaching the new point to any points in the vicinity of the click.

## ***Color Options***

### *Swap Color Palettes*

*Swap Color Palettes* toggles the displayed color palette between the default color palette and custom color palette.

### *255 Color Display*

*255 Color Display* toggles the range of the color RGB values between the default [0;1] range and the [0;255] range familiar to the Windows users. The color display range changes only the color RGB display in the HMI Configurator. The run-time color RGB values and the values saved in the drawing still use the default [0;1] range.

## ***Dynamics Options***

### *Treat Dynamics as Global on Copy*

*Treat Dynamics as Global on Copy* controls the way the dynamics is copied. If the option is set, the dynamics' parameters such as color and threshold lists are treated as global when the object the dynamics is attached to is copied and may be changed in one place for all object copies. The option also controls copying dynamics using the dynamics' *Copy* and *Paste Dynamics* options.

### *Advanced Dynamics Display*

*Advance Dynamics Display* controls the way dynamics are displayed. If the option is not checked (default), the *Edit Dynamics* dialog displays public properties of predefined dynamics for convenient editing. If the option is checked, the dialog displays all attributes of the dynamics for advanced editing. By default, this option is disabled in the *glg\_hmi\_config* file.

Refer to the *Transformation Object* chapter on page 184 of the *GLG User's Guide and Builder Reference Manual* for information on dynamics' attributes.

### **Data Browser Options**

#### *Browse Tag Names*

#### *Browse Tag Sources*

Control the usage of the Data Browser. If *Browse Tag Sources* is selected, the tag string selected in the Data Browser is assigned to the *TagSource* attribute of the tag being edited, otherwise it is assigned to the tag's *TagName* attribute.

### **Appearance**

#### *Scrollbars*

*Scrollbars* shows or hides the scrollbars in the GLG HMI Configurator window.

#### *Toolbar*

*Toolbar* shows or hides the toolbar at the top of the GLG HMI Configurator window.

### **Force File Extensions**

If checked, automatically adds ".g" extension when saving drawings if a specified filename has no extension.

### **Display OpenGL Info**

This option displays diagnostic information about the status of the OpenGL driver. If the driver has been successfully initialized, it also displays the OpenGL renderer and vendor information to assist in troubleshooting the OpenGL driver setup problems.

The *-verbose* command-line option may be used to display extended diagnostic information on Linux/Unix, or to record the information in the *glg\_error.log* file on Windows.

### **Help**

#### *Online Reference*

On Windows, *Online Reference* starts a browser with the GLG Online Documentation page. On Linux/Unix, it displays a dialog with information on how to access GLG Documentation.

### *About GLG Toolkit*

*About GLG Toolkit* displays a dialog with the GLG Toolkit version information.

255 Color Display, 84

## A

Alarm Label, 44  
 Alarms, 78  
 aligning objects, 81  
 Anchor, 24  
 Anchoring, 26  
 animation, 80  
 arc object, 23, 74  
 Arrow Type, 30  
 Arrowhead, 30  
 ArrowShape, 30  
 Attribute Properties dialog, 18  
 axis, 81

## B

BMP image, 24  
 Box Attributes object, 26

## C

chord arc object, 23  
 color RGB  
   changing range, 84  
 command actions, 79  
 CONICAL gradient, 29  
 constraints  
   tracing, 83  
 control points  
   arc object, 23  
   circle, 23, 24  
 copying  
   object, 60  
 copying with an offset, 60  
 curve rendering, 27, 76  
 custom color palette, 84  
 Custom Objects  
   adding, 59

## D

deleting  
   object, 60  
 Deleting rendering, 30  
 Direction, 26  
 Direction of a connector, 28  
 drawing

saving, 54, 55, 82

## E

EdgeType, 28  
 Edit Toolbox, 20  
 ellipse object, 75  
 ELLIPTICAL gradient, 29  
 Enabled, 44  
 EndAngle, 23  
 Export Strings, 57  
 Export Tags, 57

## F

FillAmount, 30  
 FillDirection, 29  
 flow transformation, 36  
 FontSize, 26  
 FontType, 26  
 free style shape, 76

## G

GIF image, 24  
 GIS object, 77  
 GISAngle, 28  
 GISCenter, 28  
 GISDataFile, 28  
 GISDisabled, 28  
 GISExtent, 28  
 GISLayers, 28  
 GISMapServerURL, 28  
 GISProjection, 28  
 GISStretch, 28  
 Gradient Center, 29  
 GradientAngle, 29  
 GradientColor, 29  
 GradientLength, 29  
 GradientType, 29  
 grid, aligning objects with, 81

## H

High, 45  
 High High, 45  
 HORIZONTAL connector, 28  
 HORIZONTAL text, 26

**I**

image formats, 24  
image save formats, 82  
ImageFile, 24  
ImageType, 24  
Import Strings, 57  
Import Tags, 57

**J**

JPEG image, 24

**L**

LINE WIDTH gradient, 29  
LineAnchoring, 26  
LINEAR gradient, 29  
Loading a palette, 59  
Low, 45  
Low Low, 45

**M**

marker object, 27, 76  
MarkerSize, 27  
MarkerType, 27  
MinFontSize, 26

**O**

Object Properties dialog, 17  
OpenGL diagnostics and renderer information, 85  
OpenGL driver troubleshooting, 85

**P**

Palette Description File, 59  
panning view area, 63  
parallelogram object, 74  
performance  
    measuring update performance, 81  
pie, pumpkin, 23  
PNG image, 24  
polygon object, 73, 76  
predefined dynamics, 34  
Properties dialog, 17, 18

**R**

Radius, 23  
range alarm, 45  
Read Palette, 59  
Rendering object, 29  
reset scaling transformation, 62  
Resolution, 23, 27  
rounded rectangle object, 74

**S**

saving  
    drawing, 54, 82  
scroll by dragging the drawing with the mouse, 64  
scrolling, 31  
sector arc object, 23  
shadow transparency, 29  
ShadowColor, 29  
ShadowOffset, 29  
SPHERICAL gradient, 29  
spline, 27  
SplineResolution, 27  
SplineType, 27  
StartAngle, 23  
Swap Color Palettes, 84

**T**

Tags, 78  
tags  
    tracing, 83  
Text Box, 26  
text object, 25, 76  
TextColor, 25  
TextScaling, 26  
TextString, 25  
TextType, 26  
Tooltip, 79  
tracing constraints, 83  
tracing tags, 83  
TransparentColor, 24  
trasparency, 21

**U**

Undo, 16, 62

**V**

VERTICAL connector, 28  
VERTICAL text, 26  
view  
    panning, 63  
    zooming, 63  
viewport object  
    editing focus, 64