

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *AIM APU429-16* at our website: *Click HERE*

3.4.2.3 Important slots

- void setValue(double)

  This slot sets the value of the gauge if the passed value is within the range defined by the properties *min* and *max*

3.4.3 Bulb



Figure 74: Bulb widget

Two states label stack widget. Depending on the state the bulb can turn "on" or "off". For an overview of the signals and slots see the Label Stack widget section.

3.4.3.1 Default Interfaces

- On

  Indicates the current state of the bulb

  Members:

  - Property: On (Type: bool)
  - Slot: void Switch (bool)

- Set Current Index

  Index of the current displayed label

  Members:

  - Property: currentIndex (Type: int)

3.4.3.2 Important properties

- On: bool

  This property indicates whether bulb is on or off

3.4.3.3 Important signals

- void indexChanged(int)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This signal is emitted if the current state has changed

3.4.3.4 Important signals

- void SetOn()

  Turns the bulb on

- void SetOff()

  Turns the bulb on

- void Switch()

  Toggles the bulb state

3.4.4 Calendar



Figure 75: Calendar widget

The Calendar widget provides a monthly based calendar widget allowing the user to select a date

3.4.4.1 Default Interfaces

- selectedDate

  Members:

  - Property: selectedDate (Type: QDate)
  - Signal: void activated (QDate)
  - Slot: void setSelectedDate (QDate)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.4.2 Important properties

- selectedDate: string

  This property holds the date of the calendar. The date is set via a simple string, where the year, month and day is separated via a '-'. e.g. "2007-04-30"

3.4.4.3 Important signals

- void selectionChanged()

  This signal is emitted whenever the current date has changed

3.4.5 Counter Display

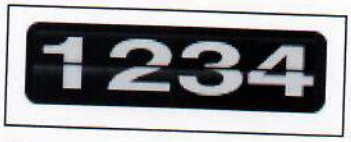


Figure 76: Counter widget

This optional Counter widget is part of the analogue widgets library (ppanawdg.dll resp. libppanawidgets.so). If the component is enabled, this widget provides a counter widget as shown in the figure above.

3.4.5.1 Default Interfaces

- value

  Members:

  - Property: value (Type: int)
  - Slot: void setValue(int)

3.4.5.2 Important properties

- value: int

  The value displayed by the counter widget

- digits: int

  The amount of digits provided by the counter widget

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.5.3 Important slots

- void setValue (int)

  Sets the value, the counter should display

3.4.6 LCD Display



Figure 77: LCD widget

An LCD can display a number in just about any size. It can display decimal, hexadecimal, octal or binary numbers. It is easy to connect to data sources using on of the display() slots.

3.4.6.1 Default Interfaces

- Set Value (int)

  Discrete value of the lcd

  Members:

  - Property: intValue (Type: int)
  - Slot: void display (int)

- Set Value (double)

  Members:

  - Property: value (Type: double)
  - Slot: void display (double)

3.4.6.2 Important properties

- value: int

  This is the central property of the widget

- mode: enum

  This property is used to set display type of the lcd. The types can be Hex, Dec, Oct or Bin

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.6.3 Important slots

- void display(int), void display(double), void display(string)

These slots can be used to set the value of the lcd to the passed value

3.4.7 LED



Figure 78: LED widget

The simple LED display widget can be turned on and off as well as turned from red to green. If only on and of is used, the colour of the LED can be set as well.

3.4.7.1 Default Interfaces

- on

  Members:

  - Property: on (Type: bool)
  - Slot: void SetLedOn (bool)

- Toggle Colour

  Members:

  - Property: green (Type: bool)
  - Slot: void SetGreen (bool)

3.4.7.2 Important properties

- colour: colour

  This property indicates colour of the LED which may be set in one of these formats:

  #RGB (each of R, G, and B is a single hex digit)

  #RRGGBB

  #RRRGGGBBB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#RRRRGGGGBBBB

A name from the list of colours defined in the list of SVG color keyword names (http://www.w3.org/TR/SVG/types.html#ColorKeywords) provided by the World Wide Web Consortium; for example, "steelblue" or "gainsboro". These colour names work on all platforms.

- on: bool

  This property indicates whether the LED is on or off

3.4.7.3 Important slots

- void SetOff()

  Sets the LED state to off

- void SetOn()

  Sets the LED state to on

- void Toggle()

  Toggles the LED state

3.4.8 LED Matrix



Figure 79: LED Matrix

This optional LED Matrix widget is part of the LGPL widgets library (pplgpl.dll resp. libpplgpl.so). If the component is enabled, this widget provides a LED Matrix widget as shown in the figure above.

3.4.8.1 Default Interfaces

- text

  Members:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Property: text (Type: string)

3.4.8.2 Important properties

- text: string

  The text to be displayed

- rows: int

  The amount of LED items per row

- columns

  The amount of LED items per column

3.4.9 Label



Figure 80: Label widget

A Label is used for displaying text or an image. No user interaction functionality is provided. Note that the label only can hold a text or a picture. If both are set, the last set is used.

3.4.9.1 Default Interfaces

- text

  Members:

  - Property: text (Type: QString)
  - Slot: void setText (QString)

3.4.9.2 Important properties

- text: string

  This property holds the displayed text of the label. (HTML-style tags are supported)

- picture: picture

  This property holds the picture to be displayed. The picture is specified via its path

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- scaledContents: bool

  This property specifies whether the picture should be scaled to the size of the label or not

3.4.9.3 Important slots

- void setText(string)

  This slot sets the text of the label.

- void setPicture(string)

  This slot sets the picture of the label, where the passed string is the picture path

3.4.10 Label Stack

The Label Stack is multi label widget where only the current activated label or picture is visible.

To setup the items of the Label Stack click right on the widget during template design and select "Edit items…". The Edit Label Stack editor will pop up, where the entries of the Label Stack can be modified

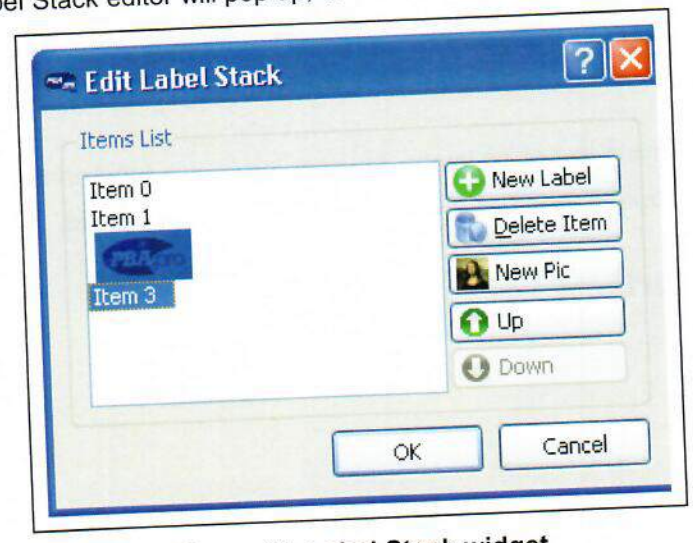


Figure 81: Label Stack widget

The properties displayed in the property editor are always valid for the current label specified by the property currentIndex. To synchronize the other labels to the properties of the current label right click on the label stack widget during design time and select one of the following menu entries:

- Synchronize geometries

  Synchronizes the geometry property of all labels with the geometry of the current label

- Synchronize all properties

  Synchronizes all properties of all labels according to the properties of the current label

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.10.1 Default Interfaces

- currentIndex

  Index of the current displayed label

  Members:

  - Property: currentIndex (Type: int)

3.4.10.2 Important properties

- currentIndex: int

  This property holds the index of the current displayed label

- States: string

  This property holds all of the states, set in the label stack. The different states are separated via '|'. (e.g. "Item 0|Item 1|Item 3|C:\pbapro\ppsplash.jpg|Item 2")

3.4.10.3 Important signals

- void IndexChanged(int)

  This signal is emitted whenever the current index has changed

3.4.10.4 Important slots

- void AddItem(String, bool, [label])

  Add a new item to the Label Stack. If no label item is passed (last optional parameter), this slot creates a new label according to the passed bool value. If the bool value is true, a text label is created, else a picture label is created. The passed string is either the text of the label or the path of the picture.

- int GetCurrentIndex()

  Gets the index of the current displayed label

- void SetCurrentIndex(int)

  Sets the index to activate the corresponding label

- int GetStatesCount()

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Gets the amount of available states

- QString GetStates()

  Returns a list of all available states, where the different states are separated via a '|'.

  (e.g. "Item 0|Item 1|Item 3|C:\pbapro\ppsplash.jpg|Item 2")

- void SetStates(string)

  Sets the available states. The different states have to be separated via a '|'.

  (e.g. "Item 0|Item 1|Item 3|C:\pbapro\ppsplash.jpg|Item 2")

- void Clear()

  Clears the whole stack

- PBLabel * GetLabel(int)

  Gets the label at the given id. NULL (for script "0x0") is returned if the given index is invalid

- PBLabel* GetCurrent();

  Gets the current label

3.4.11 Manometer



Figure 82: Manometer widget

This optional Manometer widget is part of the analogue widgets library (ppanawdg.dll resp. libppanawidgets.so). If the component is enabled, this widget provides an analogue manometer widget as shown in the figure above.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.11.1 Default Interfaces

- value

  The value to be displayed

  Members:

  - Property: value (Type: double)
  - slot: setValue (double)

3.4.11.2 Important properties

- value: double

  The value to be displayed in floating point double precision

- maximum: double

  The maximum value of the manometer

- minimum: double

  The minimum value of the manometer

- nominal: double

  The nominal value of the manometer. The range from nominal to critical will be indicated via a green scale section

- critical: double

  The critical value of the manometer. The range from critical to maximum will be indicated via a red scale section

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

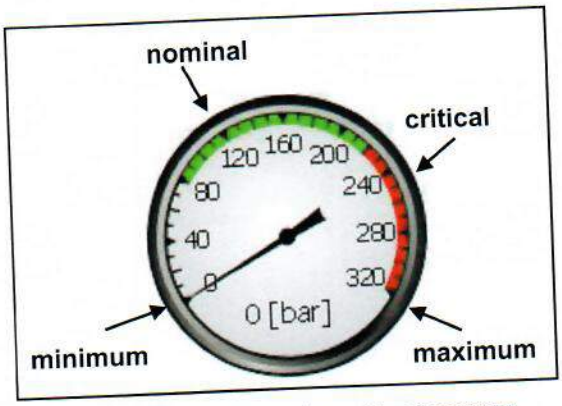


Figure 83: Manometer widget ranges

3.4.11.3 Important signals

- void valueChanged(int)

  This signal is emitted whenever the current value has changed. The passed value is in discrete representation

- void valueChanged(double)

  This signal is emitted whenever the current value has changed. The passed value is in floating point double precision

3.4.11.4 Important slots

- void setValue(int)

  The value is to be set according to a discrete value

- void setValue(double)

  The value is to be set according to a floating point double precision value

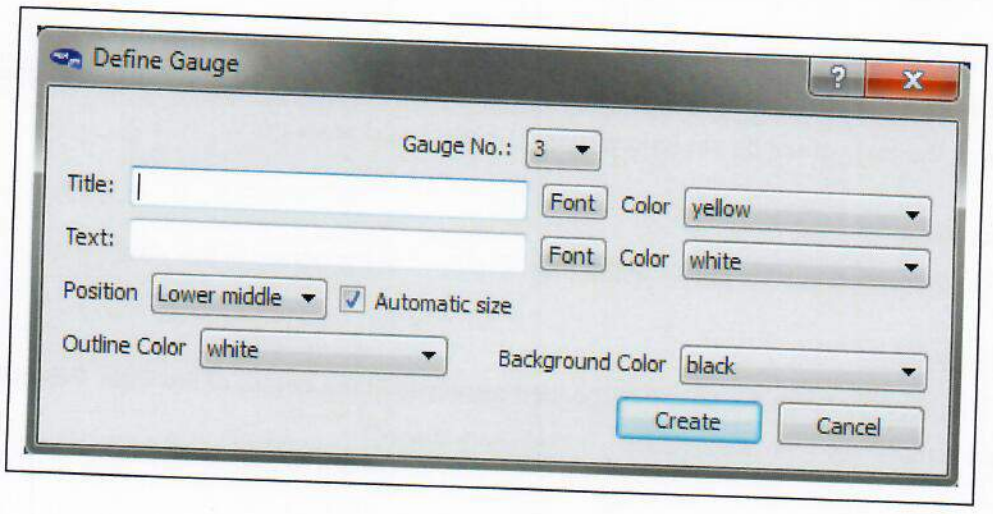
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.12 Moving Map



Figure 84: Moving map widget

This Widget allows the visualisation of the movement of an object on a map. Up to ten gauges displaying information can be contained in the display.To use this widget, a background image (the map) must be set. The property mapRealWidth holds the width of the area on the map in the same units the movement data is given. Further properties hold the coordinates of the lower right corner of the map (its south western edge) in the coordinate system the movement data is given. A gauge can be added to the widget via context menu or by double clicking the widget. This opens a dialogue, wherein all properties of the gauge can be set (c.f. fig. 79). The context menu of a gauge also offers the option to delete or modify it. Double clicking on a gauge opens a dialogue where it can be modified.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 85: Dialogue for defining gauges

A gauge consists of a coloured background with a coloured border and can display a title and a text on that background. Title and text each have their own font and colour they use. The title is displayed on top of the text and remains fixed at runtime. The text displayed by a gauge can be changed at any time by calling the appropriate slot or setting the corresponding property of the widget. Each gauge is assigned a number which defines the property and slot of the map via which its text can be accessed. The display will always be centred on the coordinates given to its properties iconXPos and iconYPos. At that position, an icon (either a circle or an arrow) can be displayed. Size, shape and colour of that icon can be set via the corresponding properties of the widget.

To create a moving display, connect the properties iconXPos and iconYPos to properties of a widget which hold your position coordinates. Alternatively the slots SetIconX(double) and SetIconY(double) can be connected to appropriate signals or the slot SetIconPos(double, double) can be called from a script. The latter slot can only be called from a script (because it takes two arguments) and is intended for use in concert with the arrow icon: It makes the arrow point in the direction of the movement – if x and y position are set separately, the arrow icon will always point in one of the four cardinal directions (since it always points in the direction of the last position change). Subsequently you can create gauges for the information you want to display (e.g. velocity) and connect the property of an object which contains that information (as text or number) to the map widget's property corresponding to the number of that gauge.

3.4.12.1 Default Interfaces

- None

3.4.12.2 Important Properties

- backgroundImage
  This property holds the map (picture) used.

- gauge<n>Text (n = 1,...10)
  These properties hold the text displayed in the corresponding gauge. If that gauge doesn't exist, the text cannot be set (and wouldn't be displayed anyway).

- iconColor
  This property holds the colour of the icon displayed in the centre of the view.

- iconShape
  This propertiy holds the shape of the icon displayed in the centre of the view. Possible shapes are "empty" (no icon is displayed), "circle" and "arrow".

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- iconSize
  This property holds the size of the icon displayed in the centre of the view.
- iconXPos
  This property holds the x (longitudinal) position the display is centred on.
- iconYPos
  This property holds the y (latitudinal) position the display is centred on.
- mapOriginLatitude
  This property holds the x (longitudinal) position of the map's lower left corner.
- mapOriginLongitude
  This property holds the y (latitudinal) position of the map's lower left corner.
- mapRealWidth
  This property holds width of the area covered by the map.

3.4.12.3 Important Signals

- None

3.4.12.4 Important Slots

- SetIconPos(double, double)
  Sets the x and y position the display is centred on at the same time.
- Zoom (double)
  Zooms the display by the given factor.

3.4.13 Progress Bar



Figure 86:Progress Bar widget

The Progress Bar widget provides a horizontal or vertical progress bar and is usually used to give the user an indication of the progress of an operation or to display that the application is still running.

3.4.13.1 Default Interfaces

- value

  Members:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Property: value (Type: int)
- Slot: void setValue (int)

3.4.13.2 Important properties

- value: int

  This is the value property of the widget. Note that the value must be within the range of the properties minimum and maximum.

- format: string

  This property holds the string used to generate the current text

  %p - is replaced by the percentage completed

  %v - is replaced by the current value

  %m - is replaced by the total number of steps

  The default value is "%p%", where the last % is the % character itself. E.g. 23%

3.4.13.3 Important signals

- void valueChanged(int)

  This signal is emitted if the value of the progress bar has changed

3.4.13.4 Important slots

- void setValue(int)

  Sets the value of the progress bar according to the passed value

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.14 SVG Gauge



Figure 87: SVG Gauge widget

The SVG (Scalable Vector Graphics) Gauge offers a gauge display in different shapes.

3.4.14.1 Default Interfaces

- value

  The value to be displayed

  Members:

  - Property: Value (Type: double)
  - slot: SetValue (double)

3.4.14.2 Important properties

- Value: double

  The value to be displayed in floating point double precision

- Maximum: double

  The maximum value of the manometer

- Minimum: double

  The minimum value of the manometer

- DisplayType: enum

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The display type of the gauge

3.4.14.3 Important signals

- void ValueChanged(double)

  This signal is emitted whenever the current value has changed. The passed value is in floating point double precision

3.4.14.4 Important slots

- void SetValue(double)

  The value is to be set according to a floating point double precision value

3.4.15 Thermometer

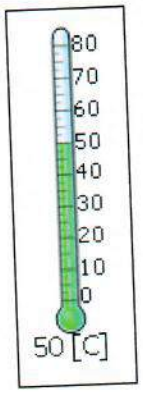


Figure 88: Thermometer widget

This optional Thermometer widget is part of the analogue widgets library (ppanawdg.dll resp. libppanawidgets.so). If the component is enabled, this widget provides a thermometer style widget as shown in the figure above.

3.4.15.1 Default Interfaces

- value

  The value to be displayed

  Members:

  - Property: value (Type: double)
  - slot: setValue (double)

3.4.15.2 Important properties

- value: double

  The value to be displayed in floating point double precision

- maximum: double

  The maximum value of the thermometer

- minimum: double

  The minimum value of the thermometer

- nominal: double

  The nominal value of the thermometer. The range from nominal to critical will be indicated via a green scale. Below the nominal the scale will be blue

- critical: double

  The critical value of the thermometer. The range from critical to maximum will be indicated via a red scale

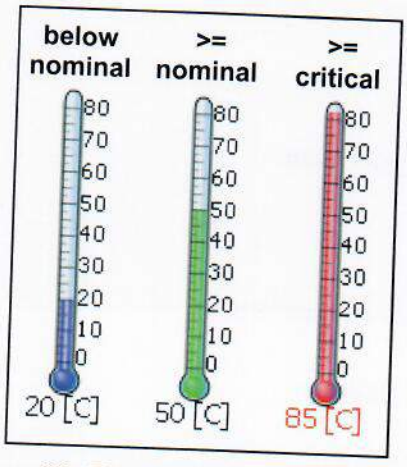


Figure 89: Thermometer state colours

3.4.15.3 Important signals

- void valueChanged(int)

  This signal is emitted whenever the current value has changed. The passed value is in discrete representation

- void valueChanged(double)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This signal is emitted whenever the current value has changed. The passed value is in floating point double precision

3.4.15.4 Important slots

- void setValue(int)

  The value is to be set according to a discrete value

- void setValue(double)

  The value is to be set according to a floating point double precision value

3.4.16 Traffic Light



Figure 90: Traffic Light widget

Three states label stack widget. Depending on the state the Traffic Light turns green (go), yellow (halt) or red (stop). See the label stack widget section for an overview of the signals and slots.

3.4.16.1 Default Interfaces

- currentState

  Indicates the current traffic light state

  Members:

  - Property: currentState (Type: State)
  - Slot: void Switch (State)

- Set Current Index

  Index of the current displayed label

  Members:

  - Property: currentIndex (Type: int)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.16.2 Important properties

- currentState: enum

  This property indicates the current state of the traffic light “go”, “halt” or “stop”

3.4.17 Vector Graphic



Figure 91: Vector Graphic widget

This widget can be used to place and display images of SVG type

3.4.17.1 Default Interfaces

- svgFile

  Path of the scalable vector graphic

  Members:

  - Property: svgFile (Type: QString)
  - Slot: void SetSvgFile (QString)

3.4.17.2 Important properties

- svgFile: string

  This property holds the path to the SVG image

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4.18 Web View



Figure 92: Web view

The Web View provides a widget that is used to view and edit web documents.

3.4.18.1 Default Interfaces

- URL

  The web content to display

  Members:

  - Property: URL (Type: QString)

3.4.18.2 Important properties

- URL: string

  This property is used to specify the web content to display

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 Paint and Decoration widgets

3.5.1 Arrow



Figure 93: Arrow Widget

This widget provides a low level Arrow item.

3.5.1.1 Important properties

- arrowWidthStart:int

  The left arrowhead width

- arrowHeightStart: int

  The left arrowhead height

- arrowWidthEnd: int

  The right arrowhead width

- arrowHeightEnd: int

  The right arrowhead height

- orientation: enum

  The orientation of the arrow, which can be set to Vertical, Horizontal, BDiagonal or FDiagonal

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.2 Ellipse and Rectangle

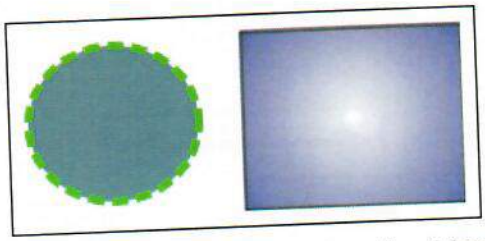


Figure 94: Ellipse and Rectangle widget

The Ellipse and Rectangle widgets can be used to visually separate regions on the dialog from others.

3.5.2.1 Important properties

- fill: bool

  Indicates whether the inner of the ellipse resp. the rectangle is filled with the colour property fillColor or if it is transparent

- fillColor: colour

  If the fill property is set to true, this property specifies the colour to fill the ellipse resp. the rectangle with

  The colour is set via one of the following strings:

  #RGB (each of R, G, and B is a single hex digit)

  #RRGGBB

  #RRRGGGBBB

  #RRRRGGGGBBBB

  A name from the list of colours defined in the list of SVG colour keyword names (http://www.w3.org/TR/SVG/types.html#ColorKeywords) provided by the World Wide Web Consortium; for example, "steelblue" or "gainsboro". These colour names work on all platforms.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- brushStyle: enum

The brush style defines how the ellipse resp. rectangle is filled, if the fill property is set to true. Supported brushStyles are:

Style	Description
NoBrush:	No brush pattern.
SolidPattern:	Uniform colour.
Dense1Pattern:	Extremely dense brush pattern.
Dense2Pattern:	Very dense brush pattern.
Dense3Pattern:	Somewhat dense brush pattern.
Dense4Pattern:	Half dense brush pattern.
Dense5Pattern:	Somewhat sparse brush pattern.
Dense6Pattern:	Very sparse brush pattern.
Dense7Pattern:	Extremely sparse brush pattern.
HorPattern:	Horizontal lines.
VerPattern:	Vertical lines.
CrossPattern:	Crossing horizontal and vertical lines.
BDiagPattern:	Backward diagonal lines.
FDiagPattern:	Forward diagonal lines.
DiagCrossPattern:	Crossing diagonal lines.
LinearGradientPattern:	Linear gradient
ConicalGradientPattern:	Conical gradient
RadialGradientPattern:	Radial gradient
TexturePattern	Texture gradient according to the texture attribute

Table 9: Supported brush styles

3.5.3 Handle



Figure 95: Handle Widgets

This widget provides a handle picture widget.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.3.1 Important properties

- Orientation: enum

  The Orientation of the handle, which can be set to Top, Right, Bottom or Left

- Type: enum

  The Handle type to be used. The available handles are placed in the folder <PBA.pro install dir>\designer\handles

3.5.4 Screws

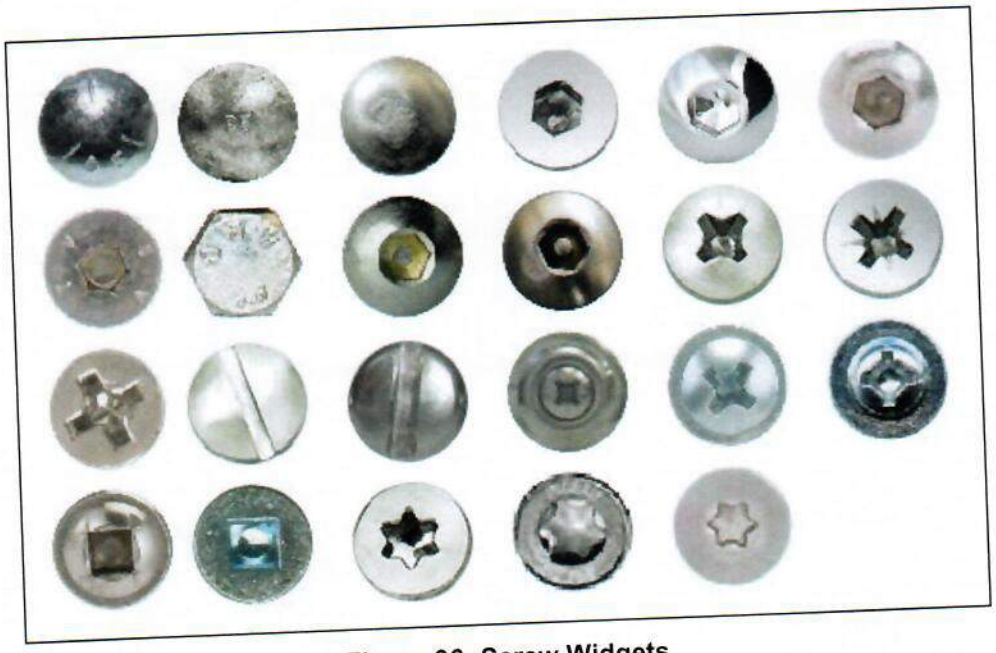


Figure 96: Screw Widgets

This widget provides a screw picture widget.

3.5.4.1 Important properties

- Rotation: int

  The rotation of the screw head in degrees between 0° and 360°

- Type: enum

  The screw head type to be used. The available screws are placed in the folder <PBA.pro install dir>\designer\screws

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6 Utility and operation widgets

3.6.1 Bool Operation



Figure 97: Bool Operation Utility widget

Utility widget used for Bool Operations

3.6.1.1 Default Interfaces

- Operand1

  First Operand

  Members:

  - Property: Operand1 (Type: bool)
  - Slot: void SetOperand2 (bool)

- Operand2

  Second Operand

  Members:

  - Property: Operand2 (Type: bool)
  - Slot: void SetOperand2 (bool)

3.6.1.2 Important properties

- Operand1: bool

  First value

- Operand2: bool

  Second value

- Operation: enum

  Operation mode, which can be one of the following:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Constant	Description
Equal	Compare to equal
NotEqual	Compare to not equal
InvertOp1	Invert Operand 1
And	Boolean AND operation
Or	Boolean OR operation
Xor	Boolean XOR operation

Table 10: Bool Operation modes

- Result: bool (read only)

  The result of the operation

3.6.1.3 Important signals

- void True()

  Emitted if the result value has become true

- void False()

  Emitted if the result value has become false

- void Value(bool)

  Emitted if the result value was set

- void ResultChanged()

  Emitted only if the result value has changed

3.6.1.4 Important slots

- void GetResult()

  Returns the result value

3.6.2 Counter



Figure 98: Counter Utility widget

The counter is a special timer, where the "count" property will be incremented on every timeout signal. In addition to the Timer's properties, signals and slots, the following is important.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.2.1 Default Interfaces

- Counter changed

  The current Counter value

  Members:

  - Property: Count (Type: int)
  - Signal: void CountChanged (int)

- Timeout

  Indicates the new Counter value

  Members:

  - Signal: void CountChanged (int)

- Run

  Starts /Stops the Timer

  Members:

  - Slot: void Run (bool)

3.6.2.2 Important properties

- MaxValue: int

  The maximum Value the counter can achieve

- Count: int (read only)

  The current count value

3.6.2.3 Important signals

- void CountChanged(int)

  This signal is emitted if the counter value has changed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.2.4 Important slots

- void Reset()

  Resets the counter property to 0

3.6.3 Display Container



Figure 99: Display container

A container widget, which can be placed on a dialog to open another display embedded to the dialog. The embedded display can be switched on the fly as well. The object of which a view is to be displayed within this container is added by drag and drop onto the display container during design time.

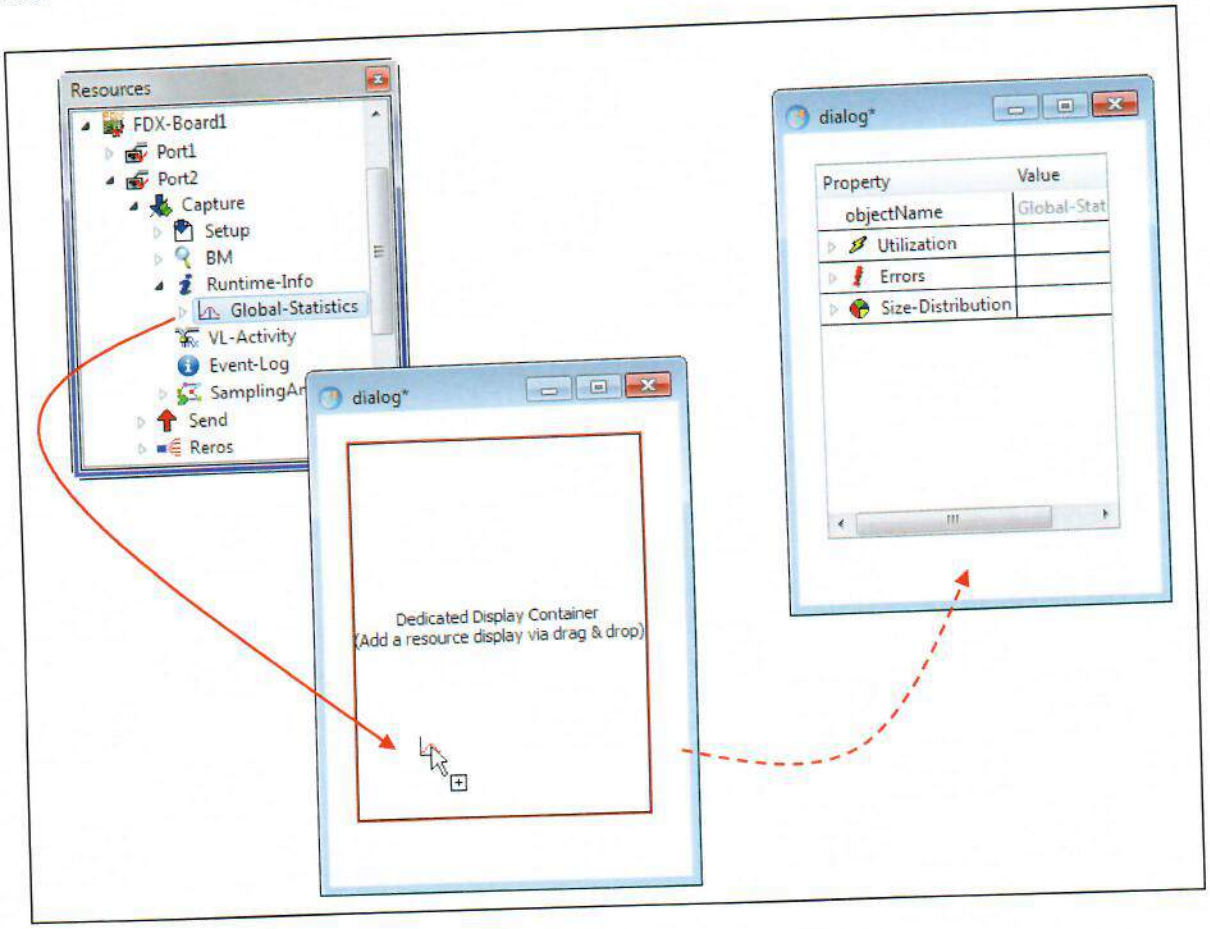


Figure 100: Display Container configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.3.1 Important properties

- ObjectPath: string

  The path of the object to open the class view of

  CustomViewName: string

- The optional class view name, if another view than the default one is to be opened

3.6.3.2 Important slots

- QWidget* OpenView(QObject *, [const QString& sViewName])

  Open the default or - if a view name given - specified view of the given object. The opened view is returned.

- QWidget* GetClassView()

  Get the opened view widget

- QWidget* OpenSelfContained()

  Open the view as self-contained and not as embedded widget

3.6.4 File Selection Widget



Figure 101: File Selection Widget

A simple file selection widget to select a file.

3.6.4.1 Important properties

- File: string

  The currently selected file

- FileTypes: string

- The file type filters which should be used by the file dialog. If multiple file type filters are used, the types have to be separated via a semicolon (;)

  Example: Images (*.png *.xpm *.jpg);Text files (*.txt);XML files (*.xml)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

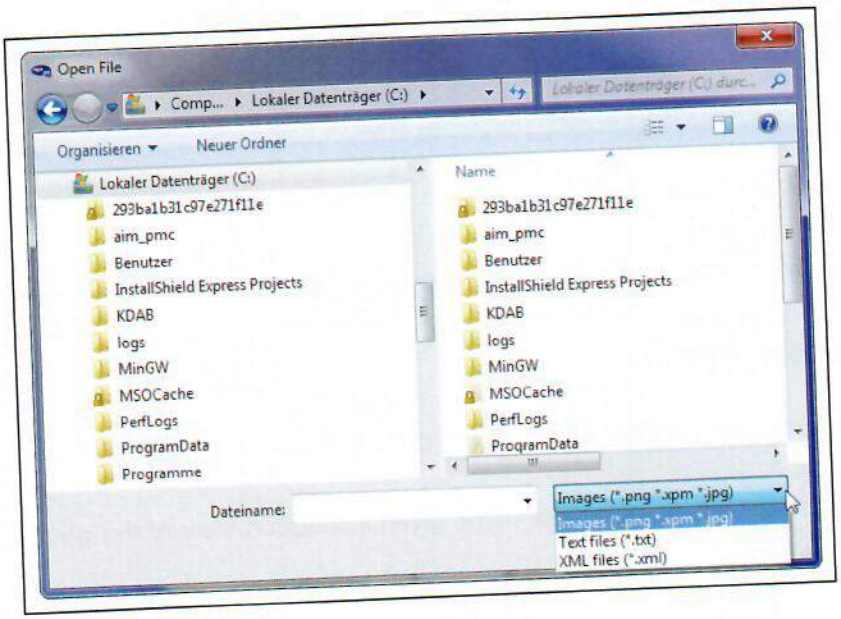


Figure 102: File Selection Widget: File Type Filters

- OpenFileDialog: bool

  Indicates if it is a file open or save dialog to be openend

- Caption: string

  The caption for the file selction dialog

3.6.4.2 Important Signals

- void FileSelected(const QString& sFile)

  This signal is emitted if a file has been selected

3.6.4.3 Important Slots

- QString SelectFile([const QString& sFile])

  Sets the current file name. If the file sFile does not exist or is not given, a file selection dialog will pop up.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.5 ID Proxy



Figure 103: ID Proxy Utility widget

Utility widget used for operations, where an additional id is needed. Usually this utility widget is needed to be able to call a slot which expects two input parameters, where on is an id.

3.6.5.1 Default Interfaces

- ValueID0 ... ValueID9

  The input value will be passed to the receiver, where the IDx is the index to be passed as second parameter value. E.g. a slot like SetValueForListEntry(int iListId,int iValue) can be accessed via the ID proxy, where the Id is the first parameter and the value the second. Setting the property ValueID0 to 20 will call the slot as follows: SetValueForList(0,20): Setting the property ValueID5 to 24 will call the slot as follows: SetValueForList(5,24)

  Members:

  - Property: ValueIdX (Type: string)

- Current Value Changed

  The current value has changed

  Members:

  - Property: CurrentvalueNum(double)
  - Property: CurrentValueStr(QString)
  - Signal: ValueUpdated(QVariant)

- ID Changed

  The current ID has changed

  Members:

  - Property: CurrentId(int)
  - Signal: IDUpdated(int)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.5.2 Important properties

- ValueID0 ... ValueID9: string

  A value placeholder to generate an id value as well if set (see interface description)

- CurrentID: int

  The currently used Id

- CurrentValueStr : string

  The currently used value in string representation

- CurrentValueNum: double

  The currently used value in floating point double precision representation

3.6.5.3 Important signals

- Void ValueUpdated(QVariant)

  The current value has changed

- Void IDUpdated(int)

  The current Id has changed

3.6.5.4 Important slots

- void SetCurrentValue(...)

  Various types of slots to set the value of the current ID

3.6.6 Latch



Figure 104: Latch Utility widget

This item can be used to send delayed signals for a stored value. E.g. write a value to the latch. If the next value occurs, the current value of the latch should be emitted before it is overwritten.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.6.1 Default Interfaces

- Latch Value

  Value of the latch (string, int or bool)

  Members:

  - Property: sValue (Type: QString)
  - Property: iValue (Type: int)
  - Property: bValue (Type: bool)
  - Slot: void SetValue (QString)
  - Slot: void SetValue (int)
  - Slot: void SetValue (bool)

- Latch Value

  Members:

  - Slot: void Latch ( )

- ValueLatched Signal

  Will be emitted if Latch is called (automatic or by user)

  Members:

  - Signal: void LatchedValue (QString)
  - Signal: void LatchedValue (int)
  - Signal: void LatchedValue (bool)
  - Signal: void LatchedValue ( )

3.6.6.2 Important properties

- bAutoLatch: bool

  If this value is set to true the signal that a new value was written is emitted immediately.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.6.3 Important signals

If the slot Latch() is called or bAutoLatch is true the various LatchedValue signals are emitted.

3.6.6.4 Important slots

- Void Latch()

  If this slot is called, the currently stored value is emitted via the LatchedValue signals.

3.6.7 Open Display



Figure 105: Open Display Utility widget

This utility widget provides a comfortable way to open another designer display or a standard view of any object. Usually this widget is combined with a button, which will activate the utility widget to open such a view.

3.6.7.1 Default Interfaces

- Open Display

  Opens the specified display

  Members:

  - Slot: QWidget* Open()

- Close Display

  Closes the specified display

  Members:

  - Slot: bool Close()

- Open/Close Display

  Opens or closes the specified display according to the given bool value

  Members:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Slot: QWidget* Open(bool)

3.6.7.2 Important properties

- Name: string

  The name of the display to open

- ObjectPath: int

  The path of the object, the display is opened of. If the OpenMode is a Designer Template the template name or path to the template is to be set here. If the OpenMode is set to Standard View, the object Path to the object, of which the display should be opened is to be specified, (e,g, "ResourceList. MIL-Board1" to open the system overview of the MIL resource named MIL-Board1).

- OpenMode: enum

  indicates the type of display to be opened:

Constant	Description
DesignerTemplate	A designer display / template is to be opened
StandardView	A standard view e.g. of a node of the of the resource tree is to be opened.

Table 11: OpenMode types of the Open View utility widget

3.6.7.3 Important slots

- QWidget* Open(bool bOpen = True)

  Opens or closes the specified display. If bOpen is set to true or not specified, the display is opened, else the display is closed.

- bool Close()

  Closes the specified display

3.6.7.4 Important signals

- void Opened(bool, QWidget*)

  This signal is emitted if a display is opened or just before the widget is closed. The bool indicates whether the display has been opened or closed, thw QWidget indicates the display.

- bool AboutToClose (QWidget*)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This signal is emitted just before the given widget is closed.

3.6.8 Timer



Figure 106: Timer Utility widget

This utility widget provides repetitive and single-shot timers.

3.6.8.1 Default Interfaces

- Timeout

  Members:

  - Signal: void timeout ( )

- Run

  Starts/ Stops the Timer

  Members:

  - Slot: void Run (bool)

3.6.8.2 Important properties

- autoStart: bool

  This property indicates whether the timer should be automatically started if the dialog is established or not

- interval: int

  This property holds the timeout interval in milliseconds

- singleShot: bool

  This property holds whether the Timer is a single-shot timer or not. A single-shot timer fires only once, non-single-shot timers fire every interval milliseconds

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.8.3 Important signals

- void timeout()

  This signal is emitted when the Timer times out

3.6.8.4 Important slots

- void Run(bool)

  Starts or stops the Timer according to the passed value

- void start([int])

  Starts the Timer. If no value is passed, the interval property is used

- void stop()

  Stops the Timer

- void restart()

  Restarts the Timer. If the Timer is not running, start() is called

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.9 User Definable Function



Figure 107: User definable Function Utility widget

This item calculates the given string in the Property Formula and returns the Result in the corresponding Property or Signal e.g. P1 + P2

3.6.9.1 Default Interfaces

- Parameter 1 - Parameter 8

  The Parameters 1 to 8 which may be used in the function. This widget provides a default interface for each parameter. Note: As all interfaces for theses 8 parameters are the same (except the Parameter number) only the first Parameter interface is described within this document.

  Members:

  - Property: P1 (Type: double)
  - Slot: void SetP1 (double)
  - Slot: void SetP1 (int)

- Result

  The Result of the Function

  Members:

  - Property: Result (Type: double)
  - Signal: void Result (double)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.9.2 Important properties

- AutoCalc: bool

  If this value is set to true the signal that a new value was written is emitted immediately

- Formula: string

  The formula of the definable function. This property specifies how the values P1 to P8 are combined with each other

  The following Operators and Functions are supported:

Operators	Meaning	Example
:=	Assign Function, which allows to define variables; P1 to P8 cannot be used as assign destinations. You may use the other properties of the function	V0:=GETVAR(V0,10)
;	Function Separator, the Result of the Function is always the last function in the sequence, Respectively the line with the RETURN (see below) Statement.	V0:=GETVAR(V0,10); V0:= V0*10;
+	Plus	P1+ P2
-	Minus	P1 – P2
*	Multiply	P1 * P2
/	Divide	P1 / P2
%	(integer) Modulo	P1%P2
^	Power function	2^3 (= 2*2*2 = 8)
Logic Operators		
==	Equal	P1 == P2
!=	Not equal	P1 != P2
<=	Less than or equal	P1 <= P2
<	Less than	P1 < P2
>=	Greater than or equal	P1 >= P2
>	Greater than	P1 > P2
&&	AND	P1 && P2
||	OR	P1 || P2
$$	XOR	P1 $& P2
Binary Operators		
&	Bitwise AND	P1 & P2
|	Bitwise OR	P1 | P2
$	Bitwise XOR	P1 $ P2
<<	Left shift	P1 << 7
>>	Right shift	P1 >> 7

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Functions	Meaning	Example
GETVAR	Get a Variable or the default value, if the Variable is not defined. It may be defined at the end of the function to have a previous result available, when the function is called the next time	V0:=GETVAR(V0,10)

RETURN	The return Statement may be used to finish the function without changing the result, by writing RETURN() at any position within the Function. If you write RETURN(<EXPR>) the expression is evaluated, and the Result Property of the function is set to the result of the expression.	V:=P1
		IF(V>10
		// No change at the result
		RETURN()
		, // Else
		// Set this value as result
		RETURN(V/10.0)
		)

SIN	Sine	SIN(P1)
COS	Cosine	COS(P1)
TAN	Tangent	TAN(P1)
ASIN	Arc sine	ASIN(P1)
ACOS	Arc cosine	ACOS(P1)
ATAN	Arc tangent	ATAN(P1)
SQRT	Square root	SQRT(P1)
LN	Natural logarithm	LN(P1)
LOG	Logarithm	LOG(P1)
ABS	Absolute value	ABS(P1)
FLOOR	Floor function	FLOOR(P1)
RAND	Random number between 0 and a value	RAND(100)
MIN	Minimum of two values	MIN(P1,P2)
MAX	Maximum of two values	MAX(P1,P2)
FMOD	Floating point modulo	FMOD(P1,P2)
AND	Same as &	AND(1,P1)
OR	Same as |	OR(1,P1)
XOR	Same as $	XOR(1,P1)
LSH	Same as <<	LSH(7,P1)
RSH	Same as >>	RSH(7,P1)
NEG	Ones complement	NEG(P1)
GETB	Get bits from the binary raw value GETB(<RAW>,<WIDTH>,<POS>) Extracts <WIDTH> Bits from the Bit Position <POS> from the <RAW> value	GETB(P1,3,2)
NOT	Logic not	NOT(P1)
GTE	Greater than or equal	GTE(P1, 5)
GT	Greater than	GT(P1, 5)
LTE	Less than or equal	LTE(P1, 5)
LT	Less than	LT(P1, 5)
EQ	Equal	EQ(P1,P2)
TST	TST(<COND>,<VALUE1>,<VALUE2>) TEST Function returns <VALUE1> if condition <COND> is true and <VALUE2> if <COND> is false.	TST(P1>10,P2,P3)
IF	Same as TST	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Misc Functions	Meaning	Example
WHILE	WHILE(<COND>,<Term>) Execute Term until COND is false. NOTE: To prevent the program to hang, the Loop is aborted after one second, if the condition has not become false	Var := P1 WHILE(Var<10 , Var:=Var + 1;)
NOP()	Does nothing, may be used e.g. in an else path of e.g. the TST function.	
PRINT(<EXP>)	This can be used to Print a Value to the Output Console. This Statement should be removed, when the development of the Function is finished.	V:= 0; PRINT(V)
CLOCK(TF)	Returns the Time since the Last Call of the Function starting with Zero TF==1 Returns Time in Milliseconds TF==2 Returns Time in Microseconds All Other Return Time in Seconds. All Return Values are double Values on Micro Second Accuracy	OT:=GETVAR(OT,0); NT:=CLOCK(0); DIFF:=NT-OT; OT:=NT;
INT	Rounds a Double Value to the Next Integer The Input may also be a String, in that case, you may specify the Number Base	INT(1.3)->1 INT("FF",16)->255
UINT	Same as INT with Unsigned Integers	
DOUBLE	Converts the Input to a double Value	
BOOL	Converts the Input to a Boolean Value	
STR	Converts the input to a string STR(Value,[<Width><Fill Char><Format Char>]) Converts the value according the format to a string <Width>: Optional the number of characters used for the conversion <Fill Char>: The character which should be used to fill e.g. leading zeros <Format Char>: The format character h / H: Hex i / d: Int (decimal) o: Octal b: Binary x: Any other Base, use the width as base s: String f : Float 1.2345 e: Float 1.123e10 g: Use e or f format, whichever is most concise	STR(123) -> "123" STR(123,h) -> "7b" STR(123,H) -> "7B" STR(123,40H)->"007B"
TIME(t,fmt)	Returns the given time t in µs time as string where the following formats are given: fmt -> Result "" -> DDDd:HHh:MMm:SS.SSSs msu -> MMm:SS.UUUUUUs dhms -> DDDd:HHh:MMm:SSs Any format string containing: %d for Day %h for Hour %m for Minute %s for Seconds %i for Milliseconds %u for Micro Seconds e.g. %h:%m:%s:%i	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Misc Functions	Meaning	Example
STRLEN	Returns the length of the string	STRLEN("Hello")->5
SUBSTR(str,start,len)	Returns the Substring starting at "start" Returns len Characters (use -1 for All until the End)	SUBSTR("abcde",1,2) -> bc
FINDSTR(str,substr,Start)	Returns the Index of the Substr from the start position	FINDSTR ("hello world","world",0) -> 6
TOUPPER	Converts the Given string to Uppercase	TOUPPER("ab")->"AB"
TOLOWER	Converts the Given string to Lowercase	TOLOWER("Ab")->"ab"
REPLACESTR (str,tofind,rep)	Replaces the tofind string in str with rep	REPLACESTR("Hello World","Hello","Hi") -> "Hi World"
INSERTSTR (str,pos,rep)	Inserts into the string at pos the rep string	INSERTSTR ("He World",2,"llo") -> "Hello World"
INCLUDE(File)	Includes the given File, which contains the function or procedure you want to use. This allows to re-use the same file in different functions. The File is searched in the following order: 1. Absolute Path 2. Relative to the Project 3. Relative to the Application Data Path (See Help About Dialog) 4. Relative to the Application Path (See Help About Dialog) 5. Relative to the Application Config Path (See Help About Dialog)	INCLUDE(C:\temp\myfunc.txt)

proc	Procedure Definition. Procedures may be defined either defined within the function itself, or you may define often used functions in the file globalfunctions.txt, which has to be located within the "Application Data Directory" (see Help About dialog). Furthermore you may define the functions in any file, which may be included with the INCLUDE statement (see above).	proc Add(a,b,c)
		(
		return(a+b+c)
		);
		
		// Now use the
		// procedure from above
		RETURN(Add(1,2,3))

Scripting Stub

Call a already sources script function:

SCRIPT(<Type>,<FunctionName>,[Param1],[Param2]...)

Calls a script function, which must be already parsed.

<Type>: The Scripter Type: Currently tcl or py is supported

<FunctionName>: The name of the function to be called.

[Param1..8]: Optional. The parameters for the script function.

The count of the parameters as well as the types have to match to the called function from the script.

e.g.

Python defines a function in the following way, the file containing the script must be sourced.

```
def TellAnswer(Value):
  return "The Answer is: " + str(Value)
```

The function above may be called with:

SCRIPT(py, TellAnswer,42)-> "The Answer is 42"

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Object Related Functions	Meaning	Example
OBJ	Returns the Object with the Given Path e.g. OBJ(dialog) returns the Root Object with the Name "dialog"	
GETPROP	Returns the Property of the Object referenced with OBJ	O:=OBJ(dialog); Val:=GETPROP(O,visible);
SETPROP	Allows to set a Property on the Object referenced with OBJ	O:=OBJ(dialog); SETPROP(O,visible,false);
PARENT	Returns the Parent Object.	O:=PARENT(dialog.chk); //returns the dialog Object
Constants	**Meaning**	**Example**
pi	The value pi (3.14159265…)	
E	The value e (2.71828183….)	
True	1	
False	0	

Table 12: Supported Function operations

- P1 to P8: double

  These properties can be used within the Formula as Input property. The Properties only accept Double Values
- V1 to V8: Variant

  These properties can be used within the Formula as Input property accepting every basic data type like strings, integer and double
- Result: double

  The result of the Formula
- VR1to VR8: Variant

  These properties can be used within the Formula as Input/ Output property accepting every basic data type. The Values may be set with:

```
VR1:=<Value>
e.g.
VR1:= "Hello World";
VR1:=9.3;
SETVR1("Hello");
RETURNVR1("Hello")
```

You may add more than one statement to a function, the single statements must be separated by semicolon. It is also possible to insert any Line Feed within the function. All signs behind a "//" are ignored and can so used as comment.

Note:

If you select the related function and press the F1 button, you get always an actual help text.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.9.3 Important signals

- void Error(string)

  This signal is emitted if an error in the formula occurred

- void Result(int), void Result(double), void ResultVR1(double) to void ResultVR4(QVariant)

  These signals are emitted if the result was calculated, respectively the related VRx Property changed

3.6.9.4 Important slots

- void Calc()

  Calling this slot forces a recalculation of the Result. *Note: This slot has to be called if AutoCalc is set to false*

- Double GetResult()

  Returns the current result value

NOTE: It is recommended to use enough brackets within the function at more complex terms. e.g. write: (P2 == 1)&&(P1 == 0) instead of P2 == 1 && P1 == 0, due to the evaluation order cannot be guaranteed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.9.5 Function If/ While Elements

The one statement in a function uses one line. To write more complex functions you may separate the statements with a semicolon.

Typically the Line Feed is ignored.

IF(Cond,<True Statements>,<False Statements>);

If your True/ False Statements are longer, you may also write

```
IF(Cond ,
    <True Statement>// this is executed when the Cond returns a value not equal 0
,
    <False Statement >// this is executed when the Cond returns a value equal 0
);
```

3.6.9.6 Usage of Variables

With the Assign Operator **:=** it is possible to create variable values, which may be used in the next execution of the function. So it is possible to use results of a previous function at another calculation cycle. **VOLD:=GETVAR(V0,1);** Sets the Variable VOLD to the value of the Variable V0 and if this does not exist, to the value 1. The variable V0 may not exist at the first execution of the function, due to it is assigned at the end of the function the first time. The ";" operator allows to separate different functions **V0:=VOLD*10** now assign the variable V0 the value of the Variable VOLD multiplied with 10. The result of this function is with each call of the "Calc" slot ten times the previous result. 1,10,100,1000

e.g. Flip Flop like Function

```
OLDVAL:=GETVAR(OLDVAL,0);
SET:=P1; // Get value for Set from P1
RESET:=P2; // Get value for Set from P2
IF((SET == 1)&&( OLDVAL == 0) ,
OLDVAL := 1; // Store the old value
RETURN(OLDVAL); // Return the value for the result
,// else
// No Operation
NOP()
);
IF( (RESET==1) && (OLDVAL==1) ,
OLDVAL:=0; // Store the Reset value
RETURN(OLDVAL) // Return it
,
NOP();
);
// No change at the Result value, RETURN() does neither modify the result nor
trigger the
// Result Signal
RETURN();
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6.10 Value Proxy



Figure 108: Value Proxy Utility widget

This utility item can be used to convert a value to another value type. This enables the user to connect two different slots to each other even if they have a different signature.

3.6.10.1 Default Interfaces

- None

3.6.10.2 Important properties

- IntValue: int

  The current value as signed integer

- BoolValue: bool

  The current value as bool value

- UintValue: uint

  The current value as unsigned integer

- DoubleValue: double

  The current value as double resp. float

- StringValue: string

  The current value as string

3.6.10.3 Important signals

If the value has changed a ValueChanged signals are emitted where the passed parameter holds the converted value.

3.6.10.4 Important slots

To set a value just call one of the various SetValue slots

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP