

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*

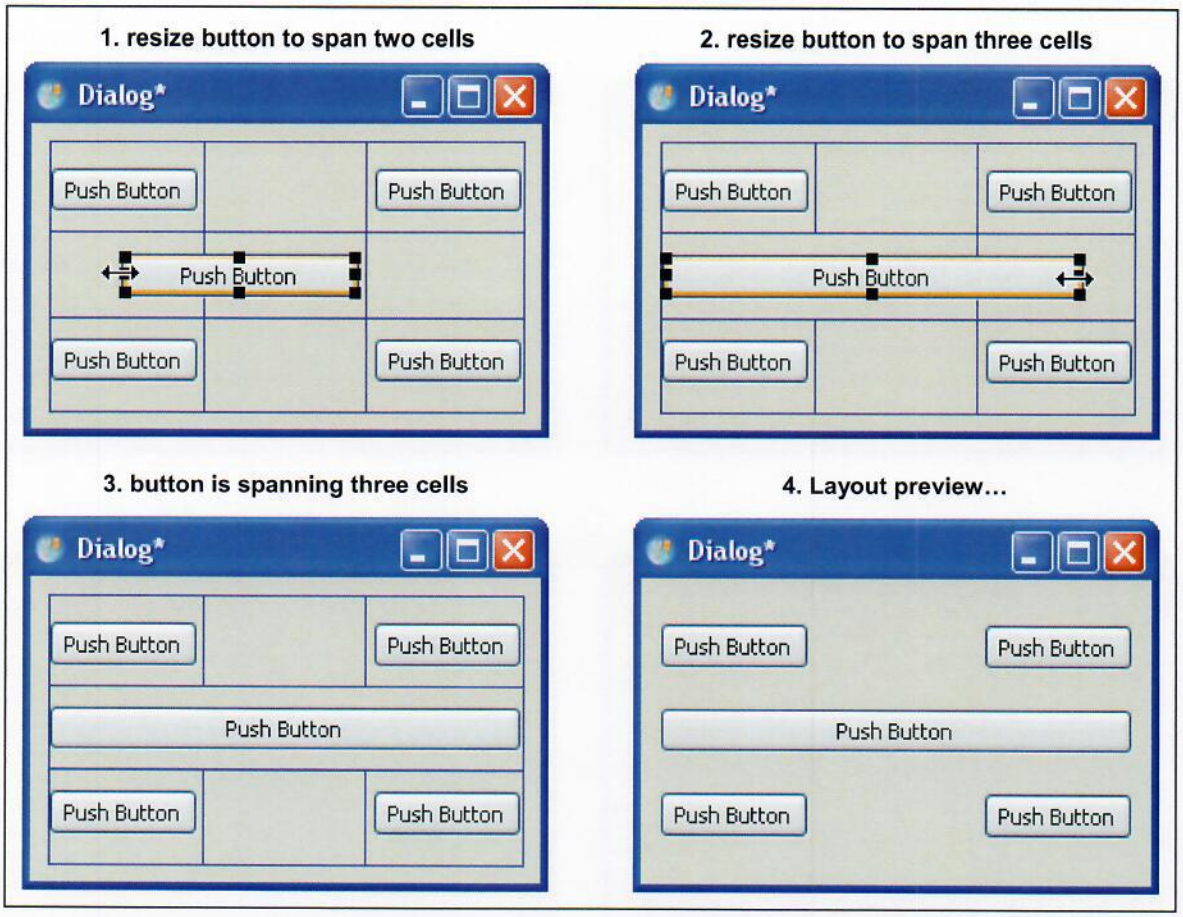


Figure 42: Spanning a widget over multiple cells

1.6.2 Spacers

In addition to the Layout itself Spacers can be added to define how the empty space respectively cells should be handled in layouts. It is recommended to define at least one widget or spacer for each column and row to ensure that the visual positioning and resizing behaviour will be the same for the design and execution mode of the template. Spacers can be added as any other widgets via drag and drop of the corresponding node from the designer tree to a cell of the layout. Depending on the configuration of the "sizePolicy" property of the added Spacer, the horizontal and vertical resizing of the cell will be laid out.

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

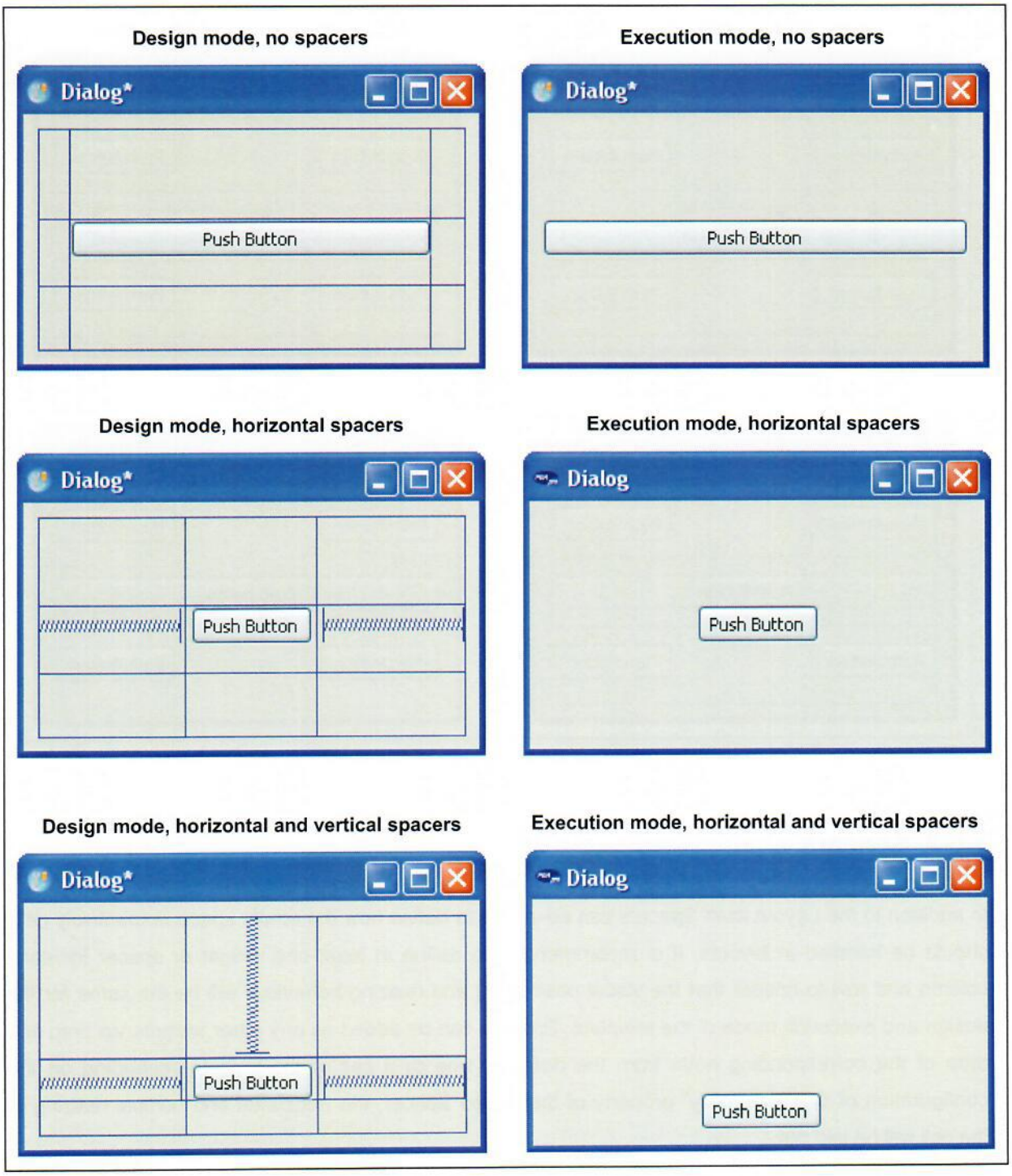


Figure 43: Spacers in Layouts

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

2 Script Level access

2.1 Dialog and Widget Creation

The Designer component offers "slots" for

- creation of "Base Containers" e.g. empty Dialogs from Panel Manager
- placing and naming of widgets on a previously created "Base Container"
- adding resources on a previously created "Base Container"

Such slots can be called from PBA.pro scripts. The Syntax of these Slots is as follows:

PanelManager:

- CreateDialog():
  - This slot is used to create a new base container
  - Returns the address of the new dialog (Base container)

Dialog:

- CreateWidget(<widget type> ,<parent>)
  - This slot is used to create a new widget on the dialog resp. on a child widget placed on the dialog
  - The type is specified via:
    - Std widgets: std.<widget> or <widget>
    - Qwt widgets: qwt.<widget>
    - Other widget libraries: <lib name>.<widget>
  - The parent is specified via its address or its path
  - Returns the widget's address (For utility widgets as timers or counters the address of the utility widget will be returned! To get the address of the timer call either `GetChildAddress` of the dialog or `GetObject` of the utility widget afterwards)
- AddResource(<path of resource>)
  - This slot is used to add a resource of the resource tree to the dialog. The resource is specified via its std. hierarchical path (e.g. "ResourceList.API-1553-Mod2.BC.Status")
  - Returns TRUE if added

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

2.2 Creation of Widget "connections" from Script Level

The Designer component offers "Slots" to define "signal-slot-" and "property connections" between the widgets placed on a dialog. The widgets resp. resources are identified via their hierarchical object path or their addresses. These two slots are to be called from the dialog itself.

- CreateSignalConnection (<sender>,
  <signal>,
  <receiver>,
  <slot>)
  - This slot is used to configure a signal-slot connection
  - The sender and receiver objects can be specified via their paths or addresses
  - If no polymorph signals resp. slots exist for the specified sender resp. receiver, the method name without parameter syntax is accepted as well (like std connect command of the script)
  - While the slot parameter types have to be the same as the signal parameter types, the slot can hold less parameters than the signal. For example a signal with the syntax *mySig (int, double)* would fit to the slots *mySlot(int, double)* and *mySlot2(int)*
  - Returns TRUE on success
- CreatePropertyConnection(<sender>,
  <property of sender>,
  <receiver>,
  <property of receiver>,
  <trigger>,
  <impulse>)
  - This slot is used to configure a property connection
  - The sender, receiver and trigger objects can be specified via their paths or addresses
  - Returns TRUE on success
- CreateInterfaceConnection(<object1>,
  <interface1>,
  <object2>,
  <interface2>)
  - This slot is used to configure a connection via the interface names
  - Object1 and object2 can be specified via their paths or addresses
  - Interface1 and interface2 are the interface names which should be used for the connection (e.g. for a Parameter *"Value as Number"*)
  - Returns TRUE on success

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

2.3 Additional slots for dialog design

PanelManager:

- GetOpenDialogs()
  - This slot is used to get a list of all open dialogs:
  - Returns a list of all opened dialogs
- GetDialogByTemplateName(<dialog name>)
  - This slot is used to find an opened template dialog. While a dialog can be accessed by its objectName directly, this method offers the possibility to access a template dialog by its template name (name displayed in the template tree)
  - Returns the dialog address. If no template was found the address will be 0. If more than one template of this type is opened, the address of the first found dialog will be returned
- LoadTemplate(<template name>)
  - This slot is used to load a template via its template name (the name displayed in the template tree)
  - Returns the address of the created dialog
- GetDesignerWidgets()
  - This slot is used to get a list of all available widgets within the designer
  - Returns a list of all widget names, where the widget types are prepended as well:
    - Standard widgets will just return their widget type: *<widget type>*
    - Qwt widgets are prepended with "qwt": *qwt.<widget type>*
    - Other widget libraries are prepended by their library extension: *<lib name>.<widget type>*
- GetChildAddress(<widget path>)
  - This slot is used to get resp. find the address of a child, placed on the dialog via its path. Especially utility widgets (the objects hold by them as timers or counters) and resources may not be found with the common script methods
  - Returns the address of the desired widget/ object

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

Dialog:

- Run()
  - This slot is used to end the design mode and run/ execute the dialog
- <scriptextension>_eval <widgetpath> objptr
  - The *..._eval* statement used with *objptr* is a standard script extension to find object addresses via their object paths. This statement can be used to find child widgets placed on a dialog as well. For example the statement `tclqt_eval dialog.pushbutton objptr` for tcl and accordingly `pypro_eval("dialog.pushbutton","objptr")` for python
  - Returns the address of the object resp. widget
- SaveTemplate() and SaveTemplateAs(<template name>)
  - These slots are used to save the current dialog as template. If `SaveTemplate()` is used, the property *templateName* is used to specify the name of the template. The call of `SaveTemplateAs` will set the template name according to the specified <template name>
  - Returns TRUE on success
- GetChildren(<type>) and GetChildrenAddresses(<type>)
  - Gets a list of all children which match the specified type, where the type is one of the `GetDesignerWidgets` list or "*resource*" to get a list of all resources placed on the dialog. To get all widgets and resources type "*all*", for widgets only type "*widgets*"
  - Returns the list of all children which match the specified type depending on the called method the children's paths or the addresses are returned
- AddLayout(<parent address>,<layout name>)
  - Adds a layout to the specified widget
  - Returns the layout as address
- GetLayout(<parent address>,<layout name>)
  - Gets the layout of the specified parent
  - Returns the layout address

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

- GetGridLayouts(<parent address>)
  - Gets all layouts of the specified widget. Note that only one layout per widget supported yet
  - Returns a list with all layout addresses added to the parent
- AddToLayout(<layout address>,<widget address>,<row>,<col>, <row span>, <col span>)
  - Adds a widget to the layout
  - The parameters `row`, `col`, `row span` and `col span` are used to specify the position in the layout. (See example script "designer-layouts.tcl" in the PBA.pro example directory how to layout widgets)
  - Returns TRUE if added
- RemoveFromLayout(<layout address>,<widget address>)
  - Removes a layout from the widget.
  - Returns TRUE if removed
- AddSpacer(<layout address>,<horizontal>,<row>, <col>,<row span>, <col span>)
  - Adds a spacer to the layout
  - Returns TRUE if added
- deleteLater()
  - To delete the dialog or one of its children the standard `deleteLater()` slot of QObject can be used. For example to delete the pushbutton "pushbutton" on a base dialog called "dialog" the following eval statement would delete the pushbutton: "tclqt_eval dialog.pushbutton member call deleteLater" or "pypro_eval("dialog.pushbutton", "member", "call", "deleteLater")"

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

2.4 Dialog creation from ASCII file

Besides the ability to create a dialog from script, the PBA.pro offers the ability to create a dialog from an ASCII file. Therefore a library script is used to parse the file and create the defined dialog. (Layouts are not supported yet).

File definition:

- The file is stored in CSV file format
- The symbol '#' as first character of a line marks the line as comment line
- To specify a resource as *sender*, *receiver* or *trigger* use its path and prepend the "*resource.*" keyword. E.g.: `resource.ResourceList.MyResource.MyObject`
- The „*#Sender*" keyword specifies the end of the widget creation resp. and the beginning of the design and connection definitions section
- Two sections are used to describe a dialog:
    - Widget creation sections (Section1)
        - 1st column: <*Widget type*>. The following properties belong to this widget until a new entry of widget type occurs
        - 2nd and 3rd columns: <*property name*> and <*value*>
        - if "*resource*" is set for column 1 the 3rd column is used to specify the resource path
    - Connection definition section (Section2)
        - 1st column: <path of sender>
        - 2nd column: <signal resp. property of sender>
        - 3rd column: <path of receiver>
        - 4th column: <slot resp. property of receiver>
        - 5th column: <*path of trigger*> (for property connections only)
        - 6th column: <*impulse*> (for property connections only)
- Currently supported property types:
    - All kind of numeric properties
    - Booleans
    - Strings
    - Size type properties

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

- Position type properties
- Rectangle type properties
- Picture, icon, bitmap or image properties via file path
- Enumerated properties via their values for the current state

Definition overview:

# (Section1) Description file for automated dialog creation					
#Widget	property	Value			
<wdg1 type1>	parent	<Par. path>			
	<property1>	<Value1>			
	<property2>	<value2>			
	…	…			
	<propertyN>	<valueN>			
<wdg2 type2>	parent	<Par. path>			
	<property1>	<Value1>			
	<property2>	<value2>			
	…	…			
	<propertyN>	<valueN>			
resource resource		<res path> <res path>			
# (Section2) Connection description					
#Sender	Signal/Property	Receiver	Slot/Property	Trigger	Impulse
<snd Path>	<signal>	<rcv path>	<slot>		
<snd Path>	<property>	<rcv path>	<property>	<trg path>	<impulse>

Table 4: CSV dialog definition overview

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

3 Widget gallery

In this section an overview of the widgets, offered by the designer, QWT and ActiveX (windows only) components are given.

Note: For all widgets the objectName of type string is a central property. The objectName is used to save the dialog structure to the template file as well as to access the widget from script level. For example: callProperty dialog.mycheckbox checked true

3.1 Containers

These widgets can be used as container widgets for other widgets (e.g. to arrange or group a certain set of widgets). The designer component offers the following container widgets:

3.1.1 Dialog

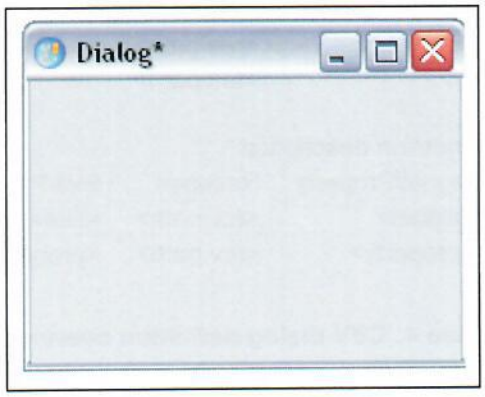


Figure 44: Base Dialog widget

This is the root widget for all template dialogs.

3.1.1.1 Default interfaces

- Established

  Emitted after the dialog is instantiated and all signals are established

  Members:

  - Signal: void Established ( )

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

- About to close

  Emitted just before the dialog is closed

  Members:

  - Signal: void AboutToBeClosed ( )

- Hide the Dialog

  Members:

  - Slot: void hide ( )

- Close the Dialog

  Members:

  - Slot: bool close ( )

3.1.1.2 Important properties

- standalone: (bool)

  This property indicates whether the dialog should be standalone or should be embedded into the PBA.pro workspace

- standaloneFrame: (bool)

  Set or remove the surrounding frame if the dialog is set as standalone dialog

- standaloneMovable: (bool)

  This property enables resp. disables the frameless move ability. By default all frameless dialogs are not movable

- syncOnRun: (bool)

  If set to true, this property forces an initial synchronization if the dialog is established. Note that only property connections can be synchronized

- Texture: (enum)

  If used the base dialog will use a background texture. The available textures are placed in the folder <PBA.pro install dir>\designer\textures

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

- templateName: (string)

  This property represents the name of the template dialog. The templates are identified via this name under the template tree node

- windowTitle: (string)

  This is the caption text of the dialog

3.1.1.3 Important signals

- void Established()

  This signal is emitted after the dialog is instantiated and all signals are established

- void AboutToBeClosed()

  This signal is emitted just before the dialog is closed

3.1.1.4 Important slots

- void Run()

  Establish all connections and run the dialog

- void SaveTemplateAs(const QString& sName):

  If a template dialog is designed via script, this slot can be used to save the template according to the given path

- QObject* GetChildAddress(const QString& sPath):

  Use this slot if you want to get the address of a child placed on the dialog. Note that the standard objptr call from script does not return the correct address for so called utility widgets and resources.

- QWidget* LoadWidgetSet(const QString& sPath, QWidget *pParent)

  Load a widget set to the specified parent container

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

3.1.2 Group Box

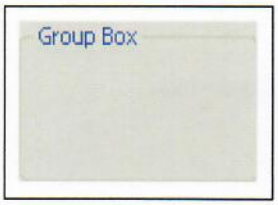


Figure 45: Group Box widget

A group box provides a frame, a title and a keyboard shortcut, and displays various other widgets inside it self. The title is on top, the keyboard shortcut moves keyboard focus to one of the group box's child widgets.

3.1.2.1 Default Interfaces

- None

3.1.2.2 Important properties

- title: (string)

  The title of the group box

3.1.3 Tab Widget

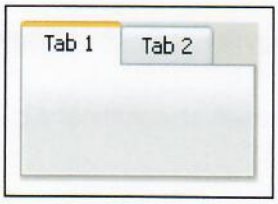


Figure 46: Tab Widget

A tab widget is a container widget and provides a stack of tabbed widgets. By default, the tab bar is shown above the page area, but different configurations are available. Each tab is associated with a different page. Only the current page is shown in the page area; all the other pages are hidden. The user can show a different page by clicking on its tab or by pressing its Alt+*letter* shortcut if it has one.

3.1.3.1 Default Interfaces

- None

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

3.1.3.2 Important properties

- numberOfTabs: int

  The amount of pages, the tab widget holds

- currentTabName: string

  The name of the current tab. This name has to be used in the hierarchy to find a child widget placed on one of the tab pages

- tabNames: string

  All available tab names, where the names are separated via a '|' (e.g. tab1|tab2)

- tabTitles: string

  All tab titles, where the titles are separated via a '|' (e.g. Tab1|Tab2)

3.1.4 Widget



Figure 47: Frame Widget

This is just a frame widget which can be used as container e.g. for widget-sets

3.1.4.1 Default Interfaces

- None

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

3.1.5 Table Widget

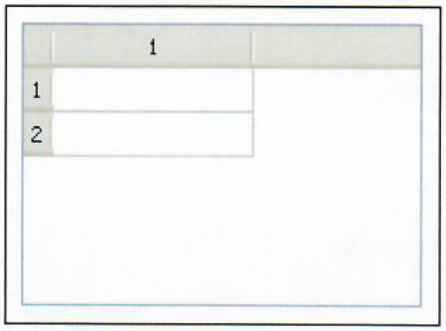


Figure 48: Table Widget

The table widget provides a graphical interface to display multiple information within a grid. Each cell can be populated with either a property of an object or a graphical widget provided by the designer. To add a widget just use the default drag and drop functionality of the designer. The widget will be added to the cell, which becomes highlighted red on dragging the widget over it. To add a resource property drag the resource over the table and drop it to the highlighted cell. A dialog will pop up to ask which property of the object should be added.

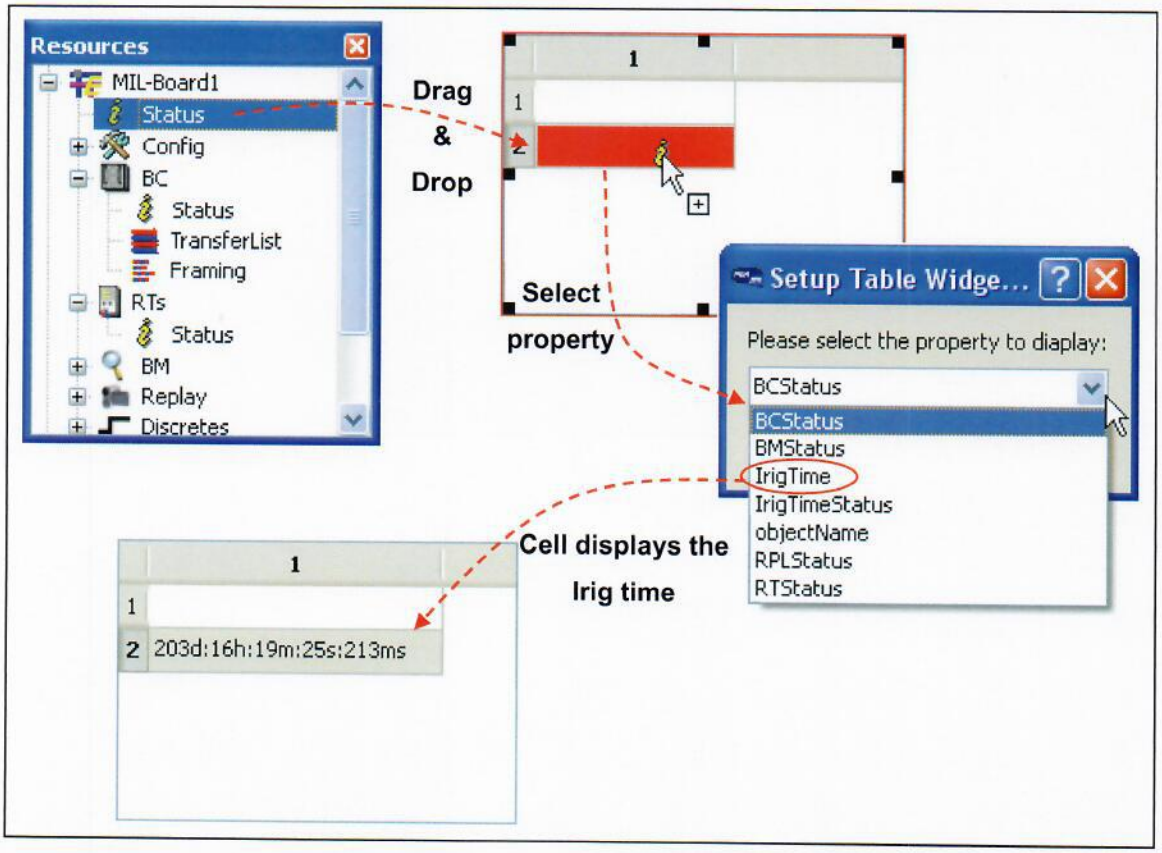


Figure 49: Adding a property to a cell

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

3.1.5.1 Default Interfaces

- None

3.1.5.2 Important properties

- columnCount: int

  The amount of available columns

- rowCount: int

  The amount of available rows

- ReadOnlyEmptyCells: bool

  Indicates whether empty cells are editable or not

- UpdateInterval: int

  The display update rate in ms for cells displaying a property of an object

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

3.2 Buttons

3.2.1 Check Box

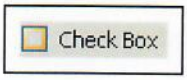


Figure 50: Check Box widget

A Check Box is an option button that can be switched on (checked) or off (unchecked). Check Boxes are typically used to represent features in an application that can be enabled or disabled without affecting others, but different types of behaviour can be implemented.

3.2.1.1 Default Interfaces

- checked

  Indicates that the check box checked state has changed

  Members:

  - Property: checked (Type: bool)
  - Signal: void toggled (bool)

- Set Checked

  Set the checked state of the check box

  Members:

  - Property: checked (Type: bool)
  - Slot: void setChecked (bool)

- Toggle

  Toggle the current checked state of the check box

  Members:

  - Slot: void toggle ( )

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

3.2.1.2 Important properties

- checked: bool

  This property holds whether the button is checked or not. Only checkable buttons can be checked. By default, the button is unchecked

- text: string

  This property holds the text shown on the button

3.2.1.3 Important signals

- void toggled (bool)

  This signal is emitted whenever a checkable button changes its state. The passed value is true if the button is checked, or false if the button is unchecked

3.2.1.4 Important slots

- void toggle()

  Toggles the state of the checkbox

- void setChecked (bool)

  Sets the state of the checkbox according to the passed value 41*

3.2.2 Push Button

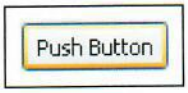


Figure 51: Push Button widget

The Push Button widget provides a command button.

The Push Button, or command button, is perhaps the most commonly used widget in any graphical user interface. E.g. push (click) a button to command the computer to perform some action, or to answer a question. Typical buttons are OK, Apply, Cancel, Close, Yes, No and Help.

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

3.2.2.1 Default Interfaces

- Clicked (Only available if the Push Button **is not** checkable)

  This signal is emitted when the button is activated (i.e. pressed down then released while the mouse cursor is inside the button)

  Members:

  - Signal: void clicked ( )

- Checked (Only available if the push button **is** checkable)

  The toggled signal is emitted whenever a checkable button changes its property *checked* is true if the button is checked; or false if the button is unchecked.

  Members:

  - Property: checked (Type: bool)
  - Signal: void toggled (bool)

3.2.2.2 Important properties

- checked: bool

  This property holds whether the button is checked or not. Only checkable buttons can be checked. By default, the button is unchecked

- text: string

  This property holds the text shown on the button

3.2.2.3 Important signals

- void clicked()

  This signal is emitted if the button was clicked (i.e. pressed down then released while the mouse cursor is inside the button)

3.2.2.4 Important slots

- void click()

  Performs a click

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

3.2.3 Radio Button



Figure 52: Radio Button Widget

The Radio Button widget provides a radio button with a text label.

A Radio Button is an option button that can be switched on (checked) or off (unchecked). Radio buttons typically present the user with a "one of many" choice. In a group of radio buttons only one radio button at a time can be checked; if the user selects another button, the previously selected button is switched off.

Radio buttons are auto-exclusive by default. If auto-exclusive is enabled, radio buttons which belong to the same parent widget behave as if they were part of the same exclusive button group.

3.2.3.1 Default Interfaces

- checked

  The toggled signal is emitted whenever a checkable button changes its property *checked* is true if the button is checked; or false if the button is unchecked

  Members:

  - Property: checked (Type: bool)
  - Signal: void toggled (bool)

3.2.3.2 Important properties

- checked: bool

  This property holds whether the button is checked or not. Only checkable buttons can be checked. By default, the button is unchecked

- text: string

  This property holds the text shown on the button

3.2.3.3 Important signals

- void toggled(bool)

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

This signal is emitted whenever a checkable button changes its state. The passed value is true if the button is checked, or false if the button is unchecked

3.2.3.4 Important slots

- void toggle()

  Toggles the state of the Radio Button

- void setChecked(bool)

  Sets the state of the Radio Button according to the passed value

3.2.4 Switch Button



Figure 53: Switch Button widget

Like every checkable button, the switch button offers two states for on and off. In contrast to a checkbox the current state is represented via the current displayed icon.

3.2.4.1 Default Interfaces

- On

  The signal is emitted if the state has changed

  Members:

  - Property: On (Type: bool)
  - Signal: void StateChanged (bool)

3.2.4.2 Important properties

- On: bool

  This property holds whether the button is on or off

- picOff: picture

  This property holds the picture displayed if the state is off. The picture is set via its path

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

- picOn: picture

  This property holds the picture displayed if the state is on. The picture is set via its path

3.2.4.3 Important signals

- void StateChanged(bool)

  This signal is emitted if the state has changed

3.2.4.4 Important slots

- void SetOn(bool)

  Sets the current button state according to the passed value

3.2.5 Toggle Button

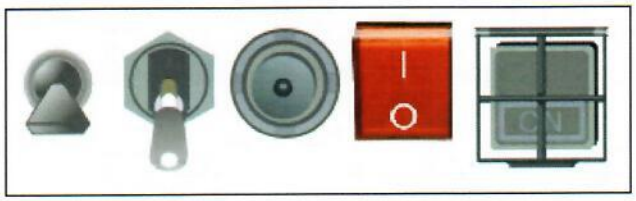


Figure 54: Toggle Button widget

Like every checkable button, the Toggle Button offers two states for on and off. In contrast to a checkbox the current state is represented via the current displayed icon.

3.2.5.1 Default Interfaces

- On

  The signal is emitted if the state has changed

  Members:

  - Property: On (Type: bool)
  - Signal: void toggled (bool)

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

3.2.5.2 Important properties

- ButtonType: enum

  The visual appearance or type of the button

Constant	OFF	ON
Standard		
RedRocker		
SecureButton		
TumblerSwitch,		
LedButton,		
Custom	Custom picture	Custom picture

Table 5: Button types of the Toggle button

- On: bool

  This property holds whether the button is on or off

- Orientation: enum

  Indicates whether the Toggle Button is to be used as vertical or horizontal toggle button

3.2.5.3 Important signals

- void toggled(bool)

  This signal is emitted if the state has changed

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

3.2.5.4 Important slots

- void SetOn(bool)

  Sets the current button state according to the passed value

3.2.6 Tool Button



Figure 55: Tool Button widget

A Tool Button is a special button that provides quick-access to specific commands or options. As opposed to a normal command button, a Tool Button usually doesn't show a text label, but shows an icon instead.

3.2.6.1 Default Interfaces

- clicked()

  Members:

  - Signal: void clicked ( )

3.2.6.2 Important properties

- text: string

  This property holds the text shown on the button

- icon: picture

  The icon to be displayed. The icon is specified via its path

3.2.6.3 Important signals

- void clicked()

  This signal is emitted when the button is activated (i.e. pressed down then released while the mouse cursor is inside the button)

3.2.6.4 Important slots

- void click()

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

Performs a click

3.3 Control widgets

The widgets placed in the input group are usually used to modify editable data like parameters

3.3.1 Combo Box

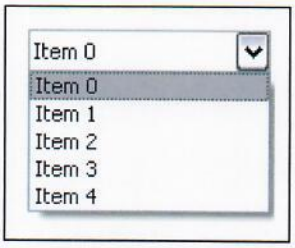


Figure 56: Combo Box widget

A Combo Box is a selection widget that displays the current item, and can pop up a list of selectable items. It may be editable, allowing the user to modify each item in the list.

The items in the combo box can hold an additional icon as well. The entries of the Combo Box are edited via the dedicated combo box editor. This editor will pop up if the user right clicks on the Combo Box during design and selects the menu entry "Edit items…".



Figure 57: Combo Box Editor

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

3.3.1.1 Default Interfaces

- Value Changed (int)

  Members:

  - Property: currentValue (Type: int)
  - Signal: void currentValueChanged (int)

- Value Changed (String)

  Members:

  - Signal: void currentIndexChanged (QString)

3.3.1.2 Important properties

- currentIndex: int

  This property holds the index of the current item in the Combo Box

- itemTitles*: string*

  Allows to setup the single entries separated by '|'. e.g. "Entry1|Entry2|Entry3"

- itemIcons: string

  Allows to setup the single icons separated by '|': e.g. "icon1.png|icon2.png|icon3.png"

3.3.1.3 Important signals

- void currentIndexChanged (int)

  This signal is sent if the current index in the combo box has changed. The passed value specifies the new index

- void currentIndexChanged (string)

  This signal is sent if the current Index in the combo box has changed. The passed value specifies the string of the new index

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

3.3.1.4 Important slots

- void setCurrentIndex(int)

  This slot sets the current index according to the passed index value. If the passed index does not exist, the selection is cleared and the current index is set to -1.

3.3.2 Date Edit, Time Edit and Date/Time Edit

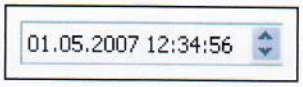


Figure 58: Date Edit, Time Edit and Date/ Time Edit widget

These widgets are used for editing dates and times.

3.3.2.1 Default Interfaces

- date (Valid for Date Edit only)

  Members:

  - Property: date (Type: QDate)
  - Signal: void dateChanged (QDate)
  - Slot: void setDate (QDate)

- time (Valid for Time Edit only)

  Members:

  - Property: time (Type: QTime)
  - Signal: void timeChanged (QTime)

- dateTime (Valid for Date / Time Edit only)

  Members:

  - Property: dateTime (Type: QDateTime)
  - Signal: void dateTimeChanged (QDateTime)
  - Slot: void setDateTime (QDateTime)

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

3.3.2.2 Important properties

- date: string (for Date Edit and Date/ Time Edit widgets)

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

  Note that the date property must be within the range of the properties maximumDate and minimumDate

- time: string (for Time Edit and Date/ Time Edit widgets)

  This property holds the date of the current widget. The date is set via a simple string, where the hour, minute and second is separated via a ':'. e.g. "12:34:56"

  Note that the time property must be within the range of the properties maximumTime and minimumTime.

- dateTime: string (for Date/ Time Edit widgets)

  This is a combined property of the properties date and time. The two strings are combined with a 'T' to specify the beginning of the time part: "2007-04-30T12:34:56"

3.3.3 Dial



Figure 59: Dial

The Dial provides a rounded range control (like a speedometer or potentiometer).

The Dial is usually used when the user needs to control a value within a program-definable range for example, with angles measured from 0 to 359 degrees. The dial can be configured as wrap around dial as well.

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

3.3.3.1 Default Interfaces

- value

  Members:

  - Property: value (Type: int)
  - Signal: void valueChanged (int)

3.3.3.2 Important properties

- value: int

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

- maximum: int

  This property is used to set the maximum value of the Dial

- minimum: int

  This property is used to set the minimum value of the Dial

- tracking: bool

  This property is used to set the Dial editable (passed value is true) or read only (passed value is false)

- wrapping: bool

  Enables resp. disables the wrap around mode of the Dial

3.3.3.3 Important signals

- void valueChanged(int)

  This signal is emitted if the value of the Dial has changed

3.3.3.4 Important slots

- void setValue(int)

  This slot sets the value of the Dial if the passed value is within the range defined by the properties minimum and maximum

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

3.3.4 Discrete Spin Box and Floating Point Spin Box



Figure 60: Spin Box widget

The Spin Box widgets allow to choose a value either by clicking the up and down buttons, by pressing Up or Down on the keyboard or by typing in the value manually to modify the currently displayed value.

3.3.4.1 Default Interfaces

- Value

  Members of Discrete Spin Box:

  - Property: value (Type: int)
  - Signal: void valueChanged (int)

  Members of Floating Point Spin Box:

  - Property: value (Type: double)
  - Signal: void valueChanged (double)

3.3.4.2 Important properties

- value: int or float

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

- maximum: int or float

  This property is used to set the maximum value of the Spin Box

- minimum: int or float

  This property is used to set the minimum value of the Spin Box

- readOnly: bool

  This property is used to set the dial editable (passed value is false) or read only (passed value is true)

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

- wrapping: bool

  Enables resp. disables the wrap around mode of the Spin Box

3.3.4.3 Important signals

- void valueChanged(int)

  This signal is emitted if the value of the Spin Box has changed

- void valueChanged(string)

  This signal is emitted if the value of the Spin Box has changed. This signal might be useful if either a suffix or a prefix is set

3.3.4.4 Important slots

- void setValue(int)

  This slot sets the value of the dial if the passed value is within the range defined by the properties minimum and maximum

3.3.5 Line Edit

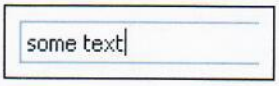


Figure 61: Line Edit widget

The Line Edit widget is a one-line text editor to enter and edit a single line of plain text

3.3.5.1 Default Interfaces

- text

  Input and output interface. If the widget is used as sender, the signal is emitted whenever the text has changed

  Members:

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

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

- Text on Return

  This signal is sent if "Return" was pressed to finish the input

  Members:

  - Signal: void ReturnPressed (QString)

3.3.5.2 Important properties

- text: string

  The text of the line edit

- readOnly: bool

  This property is used to set the line edit editable (passed value is false) or read only (passed value is true)

3.3.5.3 Important signals

- void textChanged(string)

  This signal is emitted whenever the text of the line edit has changed, where the passed value is the new text

3.3.5.4 Important slots

- void setText(string)

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

3.3.6 Object Combo Box

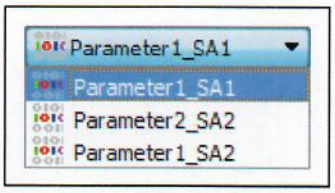


Figure 62: Object Combo Box Widget

A special combo box widget which can be configured to list child objects of a Resource. The objects to be listed need to be located below the added object (e.g. as child node in the resource tree). The depth and type of objects to be displayed are configured via the ChildSerachLevels, ClassName and ExactClassName properties. If none of the properties are configured the direct children of the added object are displayed.

To configure the combo box drag and drop a resource node onto the combo box. The node is added a parent node of which the children are to be displayed. If dedicated children should be displayed set the ClassName property to the child Type. The type is displayed in the online help (press F1 on the object). If children which are not direct but indirect children of the object are to be displayed use the ChildSerachLevels to specify the depth, the combo box should search for children.

1. Add the object via drag and drop:

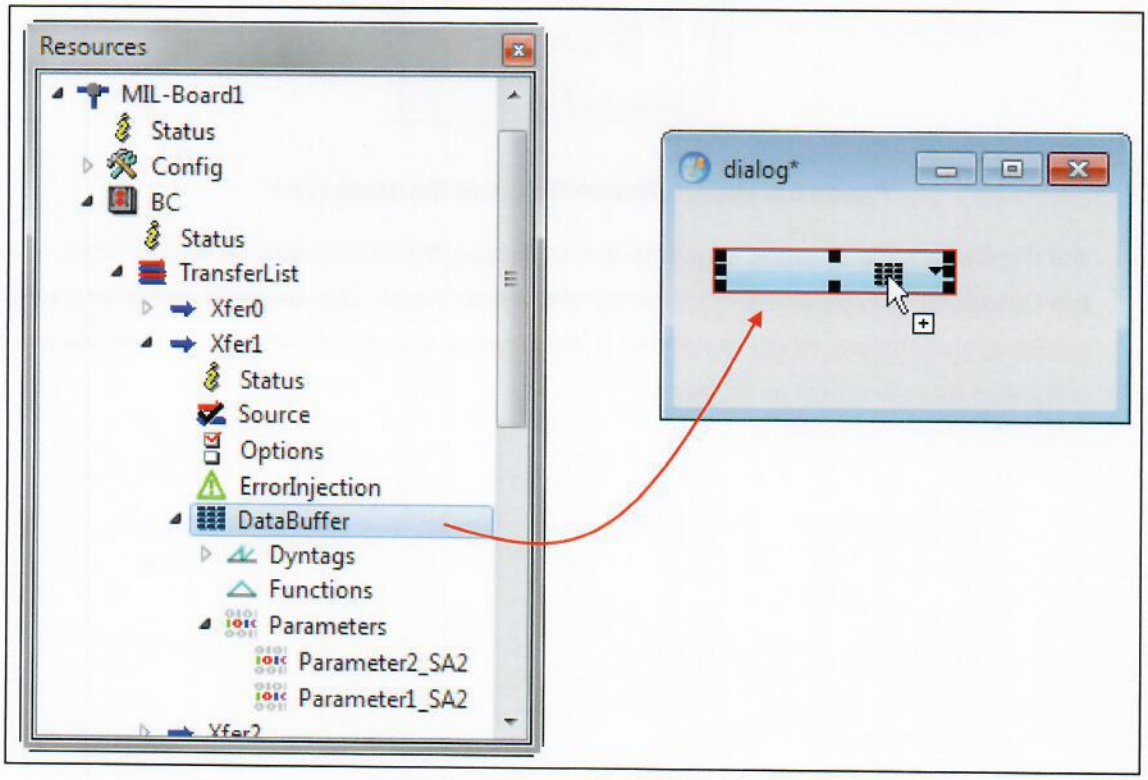


Figure 63: Object Combo Box: Add an object

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

2. If the dialog is executed, the combo box will list the direct children



Figure 64: Object Combo Box: Direct children of the added combo box

3. Press F1 on the node which should be displayed by the object combo box to get the online help of the node. In this example F1 on a parameter is pressed.

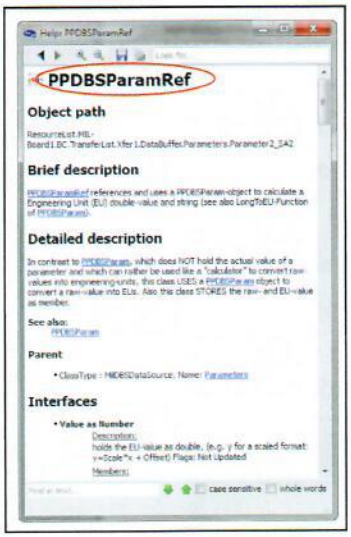


Figure 65: Object Combo Box: Get the class type

4. Set the ClassName property to specify the node type, the combo box should list items of and the ChildSearchLevels property to specify the search depth. This example demonstrates how to display parameters, which are placed 2 levels below the added buffer node. Accordingly the properties have to be set as follows:

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

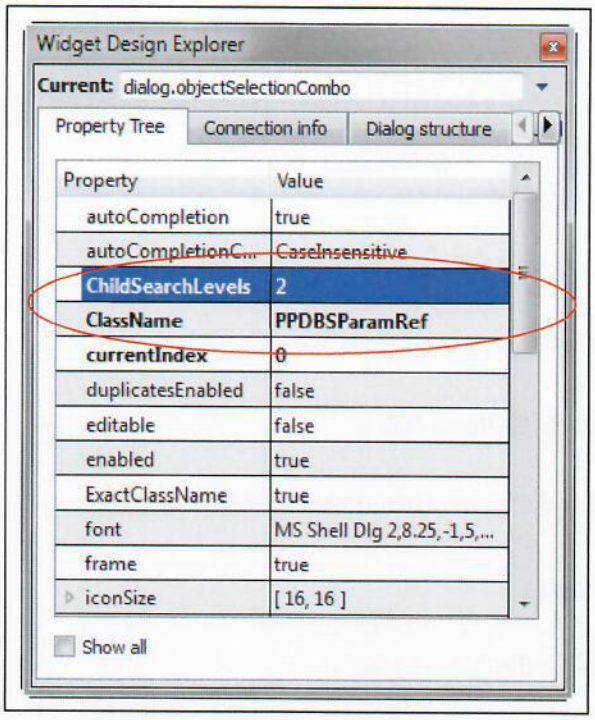


Figure 66: Object Combo Box: Configuration to display a dedicated node type

5. If the dialog is executed, the combo box will list the specified children

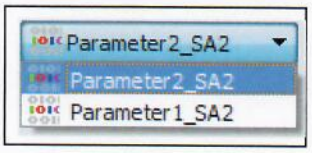


Figure 67: Object Combo Box: The specified children are listed

3.3.6.1 Important properties

- ChildSearchLevels : int

  If *ClassName* is set, this property is used to specify the maximum child search depth.

- ClassName : string

  The optional class name, the children which should be listed in the combo box have to match to.

- ExactClassName : bool

  If *ClassName* is set, this property is used to specify whether a child node must be exactly of the class name or just inherit the class.

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

- ObjectPath : string

  The path of the object of which the children should be listed by the object combo box.

3.3.6.2 Important signals

- void ObjectPathSelected (const QString &sObjectPath)

  This signal is emitted if a new child node has been selected.

- void ObjectSelected (QObject *pO)

  This signal is emitted if a new child node has been selected.

3.3.6.3 Important slots

- QString GetTextOfCurrentItem ( ) const

  Get the current item text.

- int SelectItemByText (const QString &sItem)

  Selects the first item, which matches the given item text and returns the index. If the item is not found -1 will be returned.

- void SetObject (QObject *pO)

  Change the current Parent node and reinitialize the combo box.

- void SetObjectByPath (const QString &sPath)

  Change the current Parent node and reinitialize the combo box.

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

3.3.7 Potentiometer

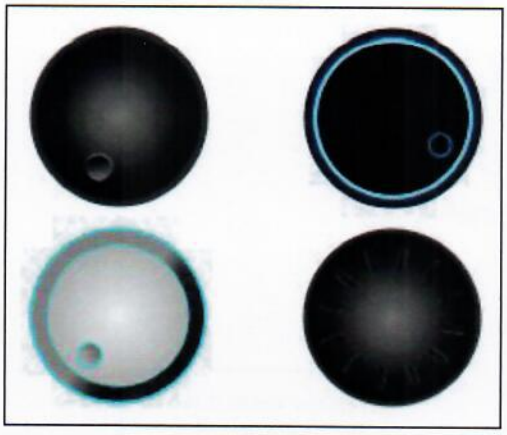


Figure 68: Line Edit widget

This optional Potentiometer widget is part of the analogue widgets library (ppanawdg.dll resp. libppanawidgets.so). If the component is enabled, this widget provides a slider like editing widget of different shapes as shown in the figure above.

3.3.7.1 Default Interfaces

- value

  Output interface to send a dedicated value to a receiver

  Members:

  - Property: value (Type: int)
  - Signal: void valueChanged(int)

3.3.7.2 Important properties

- value: int

  The value the Potentiometer

- Type: enum

  Use this option to specify the shape of the Potentiometer

3.3.7.3 Important signals

- void valueChanged(int)

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

This signal is emitted whenever the value of the Potentiometer has changed. The passed value is the new Potentiometer value

3.3.7.4 Important slots

- void setValue(int)

  Sets the value of the Potentiometer

3.3.8 Text Editor

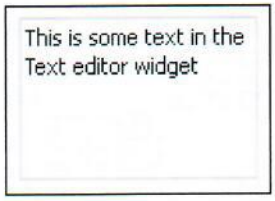


Figure 69: Text Editor widget

The Text Edit is an advanced WYSIWYG viewer/ editor supporting rich text formatting using HTML-style tags. It is optimized to handle large documents and to respond quickly to user input.

3.3.8.1 Default Interfaces

- Text changed (Plain)

  The plain text as displayed in ASCII

  Members:

  - Property: plainText (Type: QString)
  - Signal: void textChanged ( )

- Text changed (Html)

  The displayed text in html style

  Members:

  - Property: html (Type: QString)
  - Signal: void textChanged ( )

- Append Text, Prepend Date Time

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

Append the text, where the DateTimeInfoFormat property is prepended to the text

Members:

- Signal: void textChanged ( )
- Slot: void AppendWithDateTime (QString)

3.3.8.2 Important properties

- html: string

  The text of the text editor (accepts HTML-style tags)

- readOnly: bool

  This property is used to set the text edit editable (passed value is false) or read only (passed value is true)

- DateTimeInfoFormat: string

  This property is used to prepend an added text with date time information if the slot AppendWithDateTime is called. The default value of this property is "[yyyy.MM.dd hh:mm:ss] :"

  These expressions may be used for the date:

Expression	Output
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. 'Mon' to 'Sun'). Uses QDate::shortDayName().
dddd	the long localized day name (e.g. 'Qt::Monday' to 'Qt::Sunday'). Uses QDate::longDayName().
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. 'Jan' to 'Dec'). Uses QDate::shortMonthName().
MMMM	the long localized month name (e.g. 'January' to 'December'). Uses QDate::longMonthName().
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

Table 6: Date format expressions of DateTimeInfoFormat property

These expressions may be used for the time:

Expression	Output
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)

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

m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without leading zeroes (0 to 999)
zzz	the milliseconds with leading zeroes (000 to 999)
AP	use AM/PM display. AP will be replaced by either "AM" or "PM".
ap	use am/pm display. ap will be replaced by either "am" or "pm".

Table 7: Time format expressions of DateTimeInfoFormat property

All other input characters will be treated as text and not be used as an expression. Example format strings (assumed that the DateTime is 21 May 2001 14:13:09):

Format	Result
dd.MM.yyyy	21.05.2001
ddd MMMM d yy	Tue May 21 01
hh:mm:ss.zzz	14:13:09.042
h:m:s ap	2:13:9 pm
[yyyy.MM.dd hh:mm:ss] :	[2001.05.21 14:13:09] :

Table 8: Example using the date and time expressions

3.3.8.3 Important signals

- void textChanged()

  This signal is emitted whenever the text of the line edit has changed

3.3.8.4 Important slots

- void append(string)

  Appends the passed text to the current text of the text editor

- void insertHtml(string)

  Inserts the passed text at the current cursor position

- void insertPlainText(string)

  Inserts the passed text at the current cursor position

- void setText(string)

  Sets the text of the text editor to the passed text

- void appendWithDatetime(QString)

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

Appends the passed text to the current text of the text editor and prepends the information string defined via the property DateTimeInfoFormat

3.3.9 Thumb Wheel

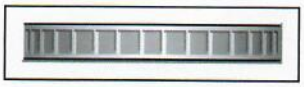


Figure 70: Thumb Wheel

The Thumb Wheel widget provides a vertical or horizontal thumb-wheel widget. The widget is often used in 2D and 3D applications to control rotation, zooming and scrolling. The rotation of the thumb wheel is translated into an integer value within a specified range.

3.3.9.1 Default Interfaces

- value

  Members:

  - Property: value (Type: int)
  - Signal: void valueChanged (int)

3.3.9.2 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

- maximum: int

  This property is used to set the maximum value of the widget

- minimum: int

  This property is used to set the minimum value of the widget

3.3.9.3 Important signals

- void valueChanged(int)

  This signal is emitted if the value of the Thumb Wheel has changed

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

3.3.9.4 Important slots

- void setValue(int)

  This slot sets the value of the Thumb Wheel if the passed value is within the range defined by the properties *minimum* and *maximum.*

3.3.10 Slider and Scroll Bar



Figure 71: Slider & Scroll Bar Widget

The Slider and Scroll Bar widgets provide a vertical or horizontal slider resp. scroll bar.

3.3.10.1 Default Interfaces

- Value

  Members:

  - Property: value (Type: int)
  - Signal: void valueChanged (int)

3.3.10.2 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

- maximum: int

  This property is used to set the maximum value of the widget

- minimum: int

  This property is used to set the minimum value of the widget

3.3.10.3 Important signals

- void valueChanged(int)

  This signal is emitted if the value of the slider or scroll bar has changed

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

3.3.10.4 Important slots

- void setValue(int)

  This slot sets the value of the slider resp. scroll bar if the passed value is within the range defined by the properties *minimum* and *maximum*

3.4 Display widgets

The widgets placed in the display group are usually used to visualise data and information of an object.

3.4.1 Analogue Clock



Figure 72: Analogue Clock widget

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

3.4.1.1 Default Interfaces

- time

  Members:

  - Property: time (Type: QTime)
  - Signal: void timeChanged(QTime)

3.4.1.2 Important properties

- time: QTime

  This property holds the current displayed time

- showCurrentDateTime: bool

  Indicates whether the current system Time (value is true) or a custom time is to be displayed

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

3.4.2 Analogue Gauge

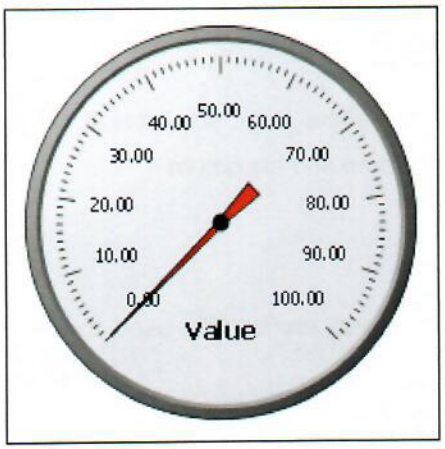


Figure 73: Analogue Gauge widget

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

3.4.2.1 Default Interfaces

- value

  Members:

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

3.4.2.2 Important properties

- value: double

  This property holds the current displayed value in floating point precision

- max: double

  This property is used to set the maximum value of the widget

- mini: double

  This property is used to set the minimum value of the widget

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