

Examples of qWC usage

Ex1 – a form with 3 buttons and 3 menu items featuring a callback that changes the background colour of the form.

Controls and properties used:

Form (Caption, Size, BCol*)

Button (Caption, Posn, Size, Event)

MenuBar

Menu, (Caption)

MenuItem (Caption, Event)

Callback functions:

cb1 – changes the background colour, uses BCol property of the form and shows backwards compatibly with rgb numbers.

Ex1w – The same as Ex1 but using the :With Control Structure

Ex2 – a form with a combo box, a button and a label. When the button is pushed the label displays the current item selected in the Combo box.

Controls and properties used:

Form (Caption, Posn, Size)

Combo (Posn, Items)

Button (Caption, Posn, Size, Event)

Label (Caption, Posn, Size)

Callback functions:

cb2 – changes the caption of the label to the selected item in the combo box

Ex3 – a form containing groups, edit boxes, label, radio buttons and push buttons. Displays the selected items/text for the Combo, Radio buttons and Edit box in the label.

Controls and properties used:

Form (Caption, Size)

Group (Caption, Posn, Size)

Button (Caption, Posn, Size, Style, Event)

Combo (Items, Posn, Size)

Edit (Text, Posn, Size)

Label* (Caption, Posn, Size)

Callback functions:

cb3 – Creates a label containing information from the Radio buttons, Combo and Edit box

Ex4 – Creates a form with a check button. The form cannot be closed without first checking the check box. Demonstrates the ability to call a callback function triggered by a change in state of the form rather than a select event on a button, in this example OnClose event. This demonstrates the same ability as ☐WC for a callback to affect the behaviour such as cancelling the close

Controls and properties used:

Form (Event)

Button (Caption, Posn, Size, Style)

Callback functions:

cb4 – Checks the state of the check box, form will close if state is 1 (i.e. the checkbox is checked)

Ex5 – A form containing two edit boxes, one of which initially displays a placeholder character, '*', instead of the entered string and check box which when checked reveals the entered string. This is meant to represent a login screen with a password section.

Controls and properties used:

Form (Caption, Size)

Label (Caption, Posn, Size)

Edit (Posn, Size, Password)

Button (Caption, Posn, Size, Style, Event)

Callback functions:

cb5 – Callback changes the 'Password' property of one of the Edit boxes, when property is set to '*' only *'s appear in Edit box

Ex6 – A form, two buttons and a label. The label displays the number stored in the Data property of the form, the buttons change or reset this number. This example is to demonstrate how to assign information to the Data property.

Controls and properties used:

Form (Caption, Size, Data)

Button (Caption, Posn, Size, Event)

Label (Caption, Posn, Size)

Callback functions:

cb6 – Callback either adds 1 to the value in the form's Data property or resets the value to 0 depending on which button is pushed.

Ex7 – A form with a treeview with three groups containing radio buttons and a checkbox. The first group contains radio buttons that toggle whether the treeview items can be edited directly in the browser. The second group contains radio buttons that toggle whether solid lines appear between the items on the treeview, this property is currently under development. The third group contains radio buttons that toggle whether checkboxes appear after the items in the treeview. The checkbox toggles whether any event messages generated by the treeview are displayed in the APL session.

Controls and properties used:

Form (Caption, Size)

Label (Caption, Posn, Size)

Button (Caption, Posn, Size, Style, Event, State)

Group (Caption, Posn, Size)

Treeview (Caption, Items, Posn, Depth, Event, Editlabels*, HasLines*, CheckBoxes*)

Callback functions:

Ex7cb – callback changes various properties, toggling on and off various options or features of the treeview.

Ex8 – A form containing a list object. There are several variations of Ex8 as listed below.

Form (Caption, Size)

List (Items, Posn, Size)

Variations

Ex8I – List object includes Index property, this property defines which item is shown on first row of the list.

Ex8S & Ex8M – List object includes style property, with Ex8S being 'Single' and Ex8M being 'multi', allowing only a single or multiple items to be selected. Multi select is currently subject to a bug report.

ExG1 – a form containing a Grid, a Combo, a Menu, three MenuItems and a MenuBar. The items in the grid can be updated using the combo box or the menus. The first 2 columns of the grid contain sample data the final column contains the sum of the first two columns, this final column is updated as the sample data is changed.

Controls and properties used:

Form (Caption, Size)

MenuBar

Menu, (Caption)

MenuItem (Caption, Event)

Combo (Items, Posn, Size, Event)

Grid (Posn, Size, RowTitles, ColTitles, CellWidths, TitleWidths, Values)

Callback functions:

Gcb1 – switches the data displayed in the first two columns of the grid between 3 preset sample data sets, then calculates the sum of the first two columns and displays the total in the third column. Works with both the combo and the menuitems.

ExG2 – a form containing a grid. The grid contains various other controls, all but Datepicker are available outside of the grid and are documented elsewhere. A standalone Datepicker control and documentation will be included in a future release. Any events triggered by interacting with the controls within the grid will display the event message in the APL session.

Controls and properties used:

Form (Posn, Size)

Grid (Posn, Size, CellWidths, Values, Input, ShowInput, CellTypes, Event)

Button (Caption, Style)

Combo (Items)

DatePicker

Edit

Callback functions:

foo2 – callback displays the event message in the APL Session

ExG3 – a form containing a grid and two groups each containing 3 radio buttons. The radio buttons can be used to change the colour of different cells within the grid.

Controls and properties used:

Form (Caption, Size)

Grid (Caption, Size, Posn, TitleWidth, CellWidths, Values, CellTypes, BCol*)

Group (Caption, Posn, Size)

Button (Caption, Posn, Size, Style, Event)

Callback functions:

Gcb3c – call back changes the BCol property of the grid. Two different colours are assigned

Gcb3p – call back changes the CellTypes property of the grid, thereby changing which cells BCol is the first colour and which Cells BCol is set to the second colour.

ExAc – a form containing a label with instructions on 2 custom accelerators, ctr+g and ctr+h. Pushing ctr+g generates a select event on the form, ctr+h generates a select event on the label.

Controls and properties used:

Form (Event, Accelerator)

Label (Caption, Size, Event, Accelerator)

Callback functions:

Afoo – displays the event message generated by the accelerators in the APL session.

ExIm – A form containing a button. When the button is selected an image of 7 random playing cards arranged in a fan is displayed. When the mouse cursor hovers over any of these cards an event message identifying the card is displayed in the APL session.

Controls and properties used:

Form (Caption, Size, Bcol)

Button (Caption, Posn, Size, Event)

Image* (Points, Size, Angle, Picture, Event)

Callback functions:

Im1 – Draws an image of 7 random playing cards arranged in a fan.

Im1c – displays the event message in the APL session for hovering the mouse over an image of a playing card.

ExH1 – a form containing a Text object, an Edit box and a Button. The text box contains two elements, the first element is defined by a fixed sample of HTML, the second element can be defined by customisable HTML displayed in the Edit box and refreshed by selecting the button.

Controls and properties used:

Form

Text (Text, TextFormat, Points)

Edit (Text, Posn, Size)

Button (Caption, Posn, Size, Event)

Callback functions:

hcb1 – callback changes the Text property of the second element of the Text object to the HTML definition in the Edit box, allowing the user to change the HTML and update the second element of the Text Object.

ExH2 – a form containing a Text object that is defined using HTML. The HTML contains qwc specific href definition `Blue`, which invokes a TextClick event when the element is clicked. The text that follows the qwc:, in this example 'bl', is the third element of the event message for the TextClick Event.

Controls and properties used:

Form (BCol*)

Text (Text, TextFormat, Points) - TextFormat is a property not found in ☐WC, if set to HTML then the Text property is interpreted as HTML, if TextFormat is set to 'Text' then the Text object displays as a text string as per ☐WC.

Callback functions:

hcb2 – the callback changes the BCol property of the form.

Charting Examples

Full documentation on Charting and Charting examples to be release soon.

ExC1 – shows between 1 and 3 data sets displayed in various different styles of graph, which data set(s) and which style of graph is used can be changed dynamically without redefining the other unchanged properties.

ExC2 – Is an example of a graph which is continuously updated from an external data source. There is a ☐Delay used in this example to make the changes less jarring to the eye.

Please note other incomplete or undocumented functions exist in this workspace which are a work in progress towards upcoming releases. Please direct any questions relating to these functions to chris@MJHSoftwareServices.co.uk

*control/property used in callback, not in the calling function.