Implement a New Property Panel

The BRAPH 2 Developers

August 17, 2024

This is the developer tutorial for implementing a new property panel. In this tutorial, you will learn how to create the generator file `*.gen.m` for a new property panel, which can then be compiled by `braph2genesis`. All property panels are (direct or indirect) extensions of the element `PanelProp`. You will use the property panel `PanelPropLogical` as an example. Then, you will be provided with an overview of example property panels using a wide range of UI objects.

Contents

Implementation of property panel (`PanelPropLogical`) 2
Example property panels for various UI objects 5

Implementation of property panel (`PanelPropLogical`*)*

To illustrate the general concepts of a property panel, you will start by implementing in detail the property panel `PanelPropLogical`, which is a direct extension of the element `PanelProp`.

Code 1: **PanelPropLogical element header.** The `header` section of the generator code in `_PanelPropLogical.gen.m` provides the general information about the `PanelPropLogical` element.

```
%% iheader!
PanelPropLogical < PanelProp (pr, panel property logical) plots the panel of
      a property logical.  (1)

%%% idescription!
PanelPropLogical plots the panel for a LOGICAL property with a checkbox.
It works for all categories. (2)

%%% ibuild!
1
```

(1) The element `PanelPropLogical` is defined as a subclass of `PanelProp`. The moniker will be `pr`.

(2) Note that more specialized property panels do not necessarily need to work for all categories.

Code 2: **PanelPropLogical element props update.** The `props_update` section of the generator code in `_PanelPropLogical.gen.m` updates the properties of the `PanelProp` element. This defines the core properties of the property panel.

```
%% iprops_update!

...

%%% iprop!
EL (data, item) is the element.
%%%% idefault!
PanelProp() (1)

%%% iprop!
PROP (data, scalar) is the property number.
%%%% idefault!
PanelProp.DRAW (2)

...
```

(1) and (2) define the default element and property for this property panel. This is necessary to ensure that the property panel refers to a property with the right format during unit testing.

Code 3: **PanelPropLogical new props.** The `props` section of the generator code in `_PanelPropLogical.gen.m` defines the user interface (UI) objects and their callbacks for the `PanelPropLogical` element.

```
%% iprops!

%%% iprop!
CHECKBOX (evanescent, handle) is the logical value checkbox. (1)
%%%% icalculate! (2)
el = pr.get('EL');
prop = pr.get('PROP');

checkbox = uicheckbox( ...
```

(1) defines the checkbox needed in the panel for a property logical. Note that this is of category EVANESCENT as it is initialized each time the code is run and is not saved.
(2) initializes the UI object.

```
  'Parent', pr.memorize('H'), ... % H = p for Panel
  'Tag', 'CHECKBOX', ...
  'Text', '', ...
  'FontSize', BRAPH2.FONTSIZE, ...
  'Tooltip', [num2str(el.getPropProp(prop)) ' ' el.getPropDescription(prop)
    ], ...
  'ValueChangedFcn', {@cb_checkbox} ...
  );

value = checkbox;
%%%% icalculate_callbacks! (3)
function cb_checkbox(~, ~)
  el = pr.get('EL'); (4)
  prop = pr.get('PROP'); (5)

  checkbox = pr.get('CHECKBOX'); (6)
  new_value = logical(get(checkbox, 'Value')); (7)

  el.set(prop, new_value)  (7)
end
```

(3) defines the callback function for when the UI object is activated.
(4) and (5) retrieve the element and property on which the callback operates.
(6) retrieves the UI object (in this case, a checkbox) and (7) the new value.
(7) writes the new value of the logical property.

Code 4: **PanelPropLogical element props update (continued).** This continues the update of the `props_update` section of the generator code in `_PanelPropLogical.gen.m`. Here, the essential properties to draw and manage the property panel are defined. Importantly, note that all these properties call the parent property calculation to ensure that the panel is correctly managed. ← Code 2

```
%% iprops_update!

...

%%% iprop!
X_DRAW (query, logical) draws the property panel. (1)
%%%% icalculate!
value = calculateValue@PanelProp(pr, PanelProp.X_DRAW, varargin{:}); % also
    warning
if value
  pr.memorize('CHECKBOX') (2)
end

%%% iprop!
DELETE (query, logical) resets the handles when the panel is deleted. (3)
%%%% icalculate!
value = calculateValue@PanelProp(pr, PanelProp.DELETE, varargin{:}); % also
    warning
if value
  pr.set('CHECKBOX', Element.getNoValue()) (4)
end

%%% iprop!
HEIGHT (gui, size) is the pixel height of the property panel. (5)
%%%% idefault!
s(4)

%%% iprop!
```

(1) draws the panel. In this case, the property panel contains only a checkbox, whose handle is memorized in (2).

(3) resets the handles when the property panel and its UI objects are deleted. In this case, it erases the handle of the checkbox in (4).

(5) specifies the height of the property panel. `s(4)` defines the height as `ceil(4 * BRAPH2.FONTSIZE * BRAPH2.S)`, where `BRAPH2.S` is by default `1`.

```
REDRAW (query, logical) resizes the property panel and repositions its
    graphical objects. (6)
%%%% !calculate!
value = calculateValue@PanelProp(pr, PanelProp.REDRAW, varargin{:}); % also
    warning
if value
  w_p = get_from_varargin(w(pr.get('H'), 'pixels'), 'Width', varargin);

  set(pr.get('CHECKBOX'), 'Position', [s(.3) s(.3) .70*w_p s(1.75)]) (7)
end

%%% !prop!
UPDATE (query, logical) updates the content and permissions of the checkbox.
     (8)
%%%% !calculate!
value = calculateValue@PanelProp(pr, PanelProp.UPDATE, varargin{:}); % also
    warning
if value
  el = pr.get('EL'); (9)
  prop = pr.get('PROP'); (10)

  switch el.getPropCategory(prop) (11)
    case Category.CONSTANT (12)
      set(pr.get('CHECKBOX'), ...
        'Value', el.get(prop), ...
        'Enable', 'off' ...
      )

    case Category.METADATA (13)
      set(pr.get('CHECKBOX'), 'Value', el.get(prop))

      if el.isLocked(prop)
        set(pr.get('CHECKBOX'), 'Enable', 'off')
      end

    case {Category.PARAMETER, Category.DATA, Category.FIGURE, Category.GUI}
      (14)
      set(pr.get('CHECKBOX'), 'Value', el.get(prop))

      prop_value = el.getr(prop);
      if el.isLocked(prop) || isa(prop_value, 'Callback')
        set(pr.get('CHECKBOX'), 'Enable', 'off')
      end

    case {Category.RESULT Category.QUERY Category.EVANESCENT} (15)
      prop_value = el.getr(prop);

      if isa(prop_value, 'NoValue')
        set(pr.get('CHECKBOX'), 'Value', el.getPropDefault(prop))
      else
        set(pr.get('CHECKBOX'), 'Value', el.get(prop))
      end

      set(pr.get('CHECKBOX'), 'Enable', 'off')
  end
end
```

(6) draws the property panel determining its graphical appearance. In this case, it just positions the checkbox in (7).

(8) updates the status of the UI objects within the panel based on the current state of the element and property to which it is linked. In this case, it just sets the value and permissions of the checkbox.

(9) and (10) retrieve the element and property to which the property panel refer.

(11) switches between the different possible property categories to make this property panel work for all of them. More specialized property panels might not need to work for all category, thus simplifying this code.

(12) When the property is a `CONSTANT`, the checkbox is disabled as it cannot be changed.

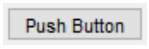
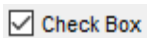
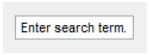
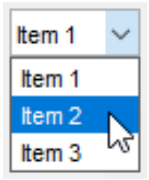
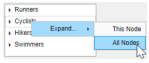
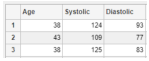
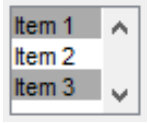
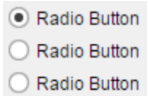
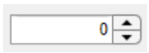
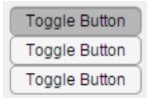
(13) When the property is a `METADATA`, the `CHECKBOX`'s enabled status depends on whether it is locked.

(14) When the property is `PARAMETER`, `DATA`, `FIGURE`, or `GUI`, the checkbox is enabled only when the property is not locked or a callback.

(15) When the property is `RESULT`, `QUERY`, or `EVANESCENT`, the checkbox is not enabled and it visualizes the default value if the property has not been calculated yet.

Example property panels for various UI objects

The implementation of `PanelPropLogical` shown in the previous section can be extended to all other user interface (UI) objects. There are several examples already available in the core code of BRAPH 2, each coupled with its corresponding property panel as an example, as shown in the table below. These can be used to guide the realization of new property panels.

UI Object	Example	Example PanelProp
uibutton		PanelPropItem
uicheckbox		PanelPropLogical
uitextarea		PanelPropStringList
uieditfield		PanelPropString
uidropdown		PanelPropOption
uicontextmenu		PanelPropMatrix
uitable		PanelPropMatrix
uilistbox		PanelPropClassList
uislider		PanelPropCell
uigauge		Not yet implemented
uiradiobutton		Not yet implemented
uispinner		Not yet implemented
uitogglebutton		Not yet implemented