

---

## EXERCISE 2

# *Action Object Method Menus*

The image shows a dialog box titled "Action Object Method". It contains three dropdown menus: "Action:" with "Show" selected, "Object:" with "Point" selected, and "Method:" with "Distance" selected. Below these is a checked checkbox for "Auto-Execute". Underneath are two text input fields labeled "Start Point List" and "End Point List". At the bottom are "Apply" and "Cancel" buttons.

### Objectives:

- Understand Action/Object/Method forms.
- Understand modularization of programming.



**Exercise Description:**

To illustrate programming an Action Object Method Menu structure, a missing branch must be added to an existing set of menus. A new class must be created to define the widgets and the callbacks for the missing branch. The branch you will be adding is *Action: Show, Object: Point, Method: Distance*. The form appears as follows:

It is okay to use the Form Builder to create the remaining class, or use one of the existing classes as a template.

**Exercise Procedure:**

1. Change directory to `ex2`. The directory contains:

**Table 1: ex2 Directory Contents**

File	Description
<i>aom.p</i>	String length definitions
<i>aom_action.pcl</i>	Action menu class
<i>aom_delete.pcl</i>	Delete option menu
<i>aom_show.pcl</i>	Show option menu

**Table 1: ex2 Directory Contents**

File	Description
<i>aom_delete_node.pcl</i>	Delete node option menu and widgets
<i>aom_delete_point.pcl</i>	Delete point option menu and widgets
<i>aom_show_node.pcl</i>	Show node option menu
<i>aom_show_node_dist.pcl</i>	Show node distance option menu and widgets
<i>aom_show_node_info.pcl</i>	Show node location option menu and widgets
<i>aom_show_point.pcl</i>	Show point option menu
<i>aom_show_point_info.pcl</i>	Show point locatoin option menu and widgets
<i>id_array.pcl</i>	Parsing routine (See Exercise ?)
<i>excr02.ses</i>	Session file to compile all functions

- You will edit one of the existing files, `aom_show_point.pcl`, and replace the following blanks:

```
***** # *****
```

This represents the distance option menu pick.

- Write a new class `aom_show_point_dist`, which manages the form. Name the file `aom_show_point_dist.pcl`. The form contains a separator, a selectframe, two point(GRID) select databoxes, another separator and an apply and a cancel button. The apply button displays the distance between the two points that the user has specified in the selectdataboxes. Use auto-execute. Display the distance in the command window as follows:

```
Point 3 and 4 are separated by dx=0.000,dy=0.00,dz=1.00
```

The cancel button should hide the form.

- Test the function by starting patran (type p3) and open a new database.
- Create some nodes and points.
- Read the session file `act_obj.ses`.
- Change the menu to Show, Point, Distance.
- Select two points and verify the distance is displayed properly.
- Press the cancel button and hide the form.

**Exercise Template:**

```

#include "appforms.p"
#include "aom.p"

CLASS aom_show_point

/*
 * This is a sample Action/Object/Method menu for the PAT306
 * class. It operates as follows:
 *
 * Action:   Show/delete
 * Object:   Point/Node
 * Method:   Location/distance (Nothing for delete action)
 *
 * Show/point/info
 *         node
 *   Presents a point selectdatabox, apply and cancel button
 *   pressing apply shows the point coordinates in the history window
 *
 * Show/Point/Distance
 *         Node
 *   Presents two point selectdataboxes, an apply and cancel button.
 *   Selecting two points and pressing the apply button indicates the
 *   distance
 *   between the two points in the history window.
 *
 * Delete/Point
 *   Presents a point selectdatabox and an apply and cancel button.
 *   Selecting a point and pressing apply deletes the point.
 */

/* Classwide Variables */
CLASSWISE WIDGET form_id, optmenu_02, item_03, item_04
CLASSWISE STRING last_method[_NUM_OBJECT_CHARS]

FUNCTION init()

    REAL yloc_main

    /*
     * Initialize variables
     */

    last_method = "aom_show_point_info"

    form_id=aom_action.get_form_id()

    yloc_main = FORM_T_MARGIN + OPT_MENU_HGT_NO_LABOVE + @
                INTER_WIDGET_SPACE+ OPT_MENU_HGT_NO_LABOVE + @
                INTER_WIDGET_SPACE

```

---

```

optmenu_02=ui_optionmenu_create(
/*  parent  */          form_id,      @
/*  callback */          "On_Method", @
/*    x     */          FORM_L_MARGIN, @
/*    y     */          yloc_main,    @
/* label_length */      0.69999999,  @
/*  label   */          "Method:",    @
/* label_above */      FALSE)

item_03=ui_item_create(
/*  parent  */          optmenu_02,   @
/*  name    */          "aom_show_point_info", @
/*  label   */          "Location",    @
/* toggleable */      TRUE)

item_04=ui_item_create(
/*  parent  */          ***** 1 ***** @
/*  name    */          ***** 2 ***** @
/*  label   */          ***** 3 ***** @
/* toggleable */      ***** 4 ***** )

END FUNCTION /* init */

FUNCTION display()

/*
 * Display the Show option menu
 */

ui_wid_set( optmenu_02, "DISPLAY", TRUE )

/*
 * Restore the previously selected object
 */

aom_show_point.On_Method( last_method )

END FUNCTION /* display */

FUNCTION exit()

ui_wid_set( optmenu_02, "DISPLAY", FALSE)
ui_exec_function( last_method, "exit")

END FUNCTION /* display */

FUNCTION On_Method(name)
/*
 * This is a callback for the optionmenu named optmenu_02
 */
STRING name[]

```

```
IF( name != last_method ) THEN ui_exec_function( last_method, "exit")
  ui_exec_function( name, "display")
  last_method = name
```

```
END FUNCTION /* On_Method */
```

```
END CLASS /* aom_show_point */
```

```
*1* optmenu_02,  
*2* "aom_show_point_dist",  
*3* "Distance",  
*4* TRUE
```

