

---



## EXERCISE 3



# *The Solution Parameters Form*



### Objectives:

- Write a PCL function which creates a form and puts a databox and buttons on it.



## Problem Description:

In this Exercise we write a PCL function which creates a form, adds a databox to input the analysis convergence tolerance and an OK and Cancel button.

## Suggested Exercise Steps:

- Write a PCL function using the online editor.
- Run the C pre-processor
- Compile the function
- Verify the PCL function

## Exercise Procedure:

Fill in the blanks

1. Either use vi or jot as the text editing tool. Open the file named *solution\_param.template*. It should already exist in your directory. Fill in the missing sections marked by:

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

2. Rename the file to *solution\_param.pcl* and compile the function.

First, since the file contains define strings to locate the widgets, we need to run the c-preprocessor (cpp) against the file. To run the c-preprocessor type:

```
>/usr/lib/cpp -P -I/patran/patran3/customization \  
solution_param.pcl solution_param.cpp
```

The file *solution\_param.cpp* should appear as result

Type **p3** at the prompt and **<return>**.

After the main menu and command window appear, type **!!input solution\_param.cpp** in the command line

Resolve any compile errors by editing **solution\_param.pcl**, running the c-preprocessor and re-compiling in *p3*.

---

3. Test the function.

The form can be displayed by calling the “init” and “display” routines as follows:

In the P3/Patran command window type:

```
solution_param.init()
solution_param.display()
```

or, you may combine the two calls using the following PCL function:

```
ui_exec_function (“solution_param”, “display”)
```

4. Verify the result:

Verify that *p3epilog.pcl* exists in your local directory which you are running p3 from. It should contain:

```
!!input training.pcl
training.init()
```

Verify that *p3prolog.pcl* exists in your local directory which you are running p3 from. It should contain the line:

```
global real convergence_tol = 1.E-3
```

Enter **p3** at the prompt.


Select **Exercise 3** from the *Training menu*.

The following form should appear:





**Exercise 3**



---

## Sample Solution

```

/*$$ Use of PCL in creating customized forms/widgets
*
* Purpose:
* Create a form with one databox, an ok and a cancel button.
* The callback for the ok button will get the string entered
* in the databoxes and write them into a global variable.
*
* Input:
* <None>
*
* Output:
* <None>
*
* Log:
*
* Notes:
* This is exercise 3 of the PAT305 class.
*
*/

***** 1 *****

CLASS solution_param

/* Variable initialization */

CLASSWIDE widget form_id, @
dbx1_id

FUNCTION INIT()

REAL y_loc, form_x_location
GLOBAL REAL convergence_toler

form_x_location = FORM_X_LOC-FORM_WID_SML - @
FORM_L_BORDER - @
FORM_R_BORDER
/*
* Create the form

```

### Exercise 3

---

```
*/

form_id=UI_FORM_CREATE( @
/*callback*/ "", @
/* x */ form_x_location, @
/* y */ FORM_Y_LOC , @
/* position */ "UL", @
***** 2 *****
/* height */ FORM_HGT_FULLL, @
/* label */ "Solution Parameters ", @
/* iconname */ "" )

y_loc = FORM_T_MARGIN

/*
* Create the "Enter the Force Amplitude data box"
*/

dbx1_id= UI_DATABOX_CREATE( @
/* parent */ form_id, @
/* callback */ "", @
***** 3 *****
***** 3 *****
/* label_length */ 0.0, @
***** 3 *****
/* label */ "Convergence Tolerance", @
/* value */ str_from_real( convergence_toleration ),@
/* label_above */ TRUE, @
/* datatype */ "REAL", @
/* num_vals */ 1 )

y_loc += DBOX_HGT_LABOVE + @
2*INTER_WIDGET_SPACE

/*
* Create the "ok" button
```

```

*/

ui_button_create( @
/* parent */ form_id, @
***** 4 *****
/* x */ BUTTON_HALF_X_LOC1, @
/* y */ y_loc, @
/* width */ BUTTON_WID_HALF, @
/* height */ 0.0, @
/* label */ "OK", @
/* labelinside */ TRUE, @
/* highlight */ TRUE )

/*
* Create the "Reset" button
*/

ui_button_create( @
/* parent */ form_id, @
/* callback */ "reset_cb", @
/* x */ BUTTON_HALF_X_LOC2, @
/* y */ y_loc, @
/* width */ BUTTON_WID_HALF, @
/* height */ 0.0, @
/* label */ "RESET", @
/* labelinside */ TRUE, @
/* highlight */ FALSE )

y_loc += BUTTON_DEFAULT_HGT + FORM_B_MARGIN

ui_wid_set( form_id, "HEIGHT", y_loc )

END FUNCTION
FUNCTION DISPLAY()

GLOBAL REAL convergence_toler

ui_wid_set( dbx1_id,"VALUE", convergence_toler )
ui_form_display( "solution_param")

```

## Exercise 3

```
END FUNCTION

FUNCTION ok_cb()

FUNCTION ok_cb()

GLOBAL REAL convergence_tolcr

***** 5 *****

***** 5 *****
***** 5 *****
***** 5 *****

END FUNCTION /* ok_cb end function */

FUNCTION reset_cb()

UI_WID_RESTORE ("solution_param")

END FUNCTION /* restore_cb end function */

END CLASS /* solution_param class */
```

---

## Solutions

```
1) #include "appforms.p"
2) /* width */ FORM_WID_SML, @
3) /* x */ UNFRAMED_L_MARGIN, @
   /* y */ Y_LOC, @
   /* label_length */ / 0.0, @
   /* box_length */ / DBOX_WID_SINGLE, @
   /* label */ "Convergence Tolerance", @
4) /* callback */ "ok_cb", @
5) UI_WID_GET (dbox_id, "VALUE", convergence_tolcr)
   UI_WID_SAVE ("solution_param")
   UI_FORM_HIDE ("solution_param")
   select_focus.exit()
```

