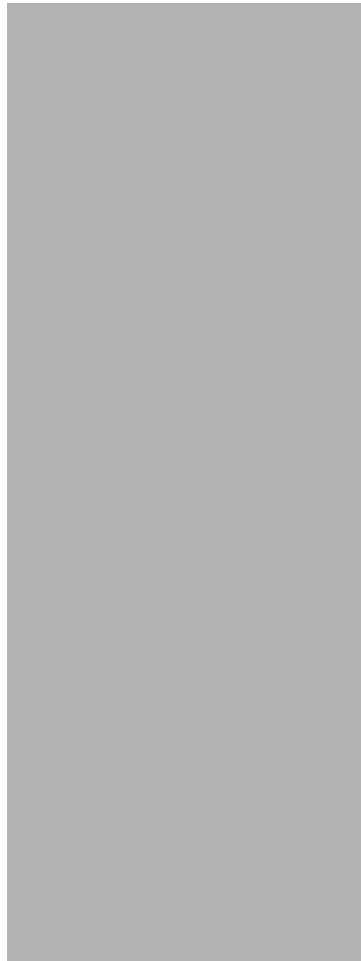



EXERCISE 9



The Analysis Form



Objectives:

- Edit *exercise_09.template* to add the callbacks for solution parameters and load case selection
- Create the **Miracle Analysis** form



Problem Description:

In this Exercise you will edit the PCL function **exercise_09.template** to put three buttons on the structural analysis form for Miracle. The button titles are:

Solution Parameters...	}	for “analyze” action
Select Load Cases...		
Select Results File...	}	for “Read Results” action

We then compile the PCL function into *miracle.plb* so we can display it.

Suggested Exercise Steps:

- Edit *exercise_09.template*. You may want to name this function *miracle_load_aom_data.pcl* to be consistent with P3/Patran conventions.
- Compile *exercise_09.pcl* into **miracle.plb**
- Compile *solution_param.pcl*, *miracle_select_file.pcl* and *miracle_loadcases.pcl* into **miracle.plb**
- Open a database and verify the analysis form callbacks.

Exercise Procedure:

1. Either use vi or jot as the text editing tool. Open **exercise_09.template**. Edit the missing lines to add three buttons and callbacks to present the forms created in exercise 3, 4, and 5. Rename the file *miracle_load_aom_data.pcl*
2. Compile *miracle_load_aom_data.pcl* into **miracle.plb**.

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

After the main menu and command window appear, type:

```
!!compile miracle_load_aom_data.pcl into miracle.plb
!!compile solution_param.pcl into miracle.plb
!!compile miracle_select_file.pcl into miracle.plb
!!compile miracle_loadcases.pcl into miracle.plb
```

Resolve any compile errors by editing *exercise_09.pcl* and re-executing the compile command.

3. Verify the callbacks.

Open database **miracle_template.db** (from exercise 8). Select the **Analysis** radio button from the *Control Panel*.

Click the **Select Load Cases...** button.

The *Loadcase Select* form should appear as below.



Click **OK**.

Select the **Solution Parameters...** button.

The solution parameters form should appear which contains the convergence tolerance button.

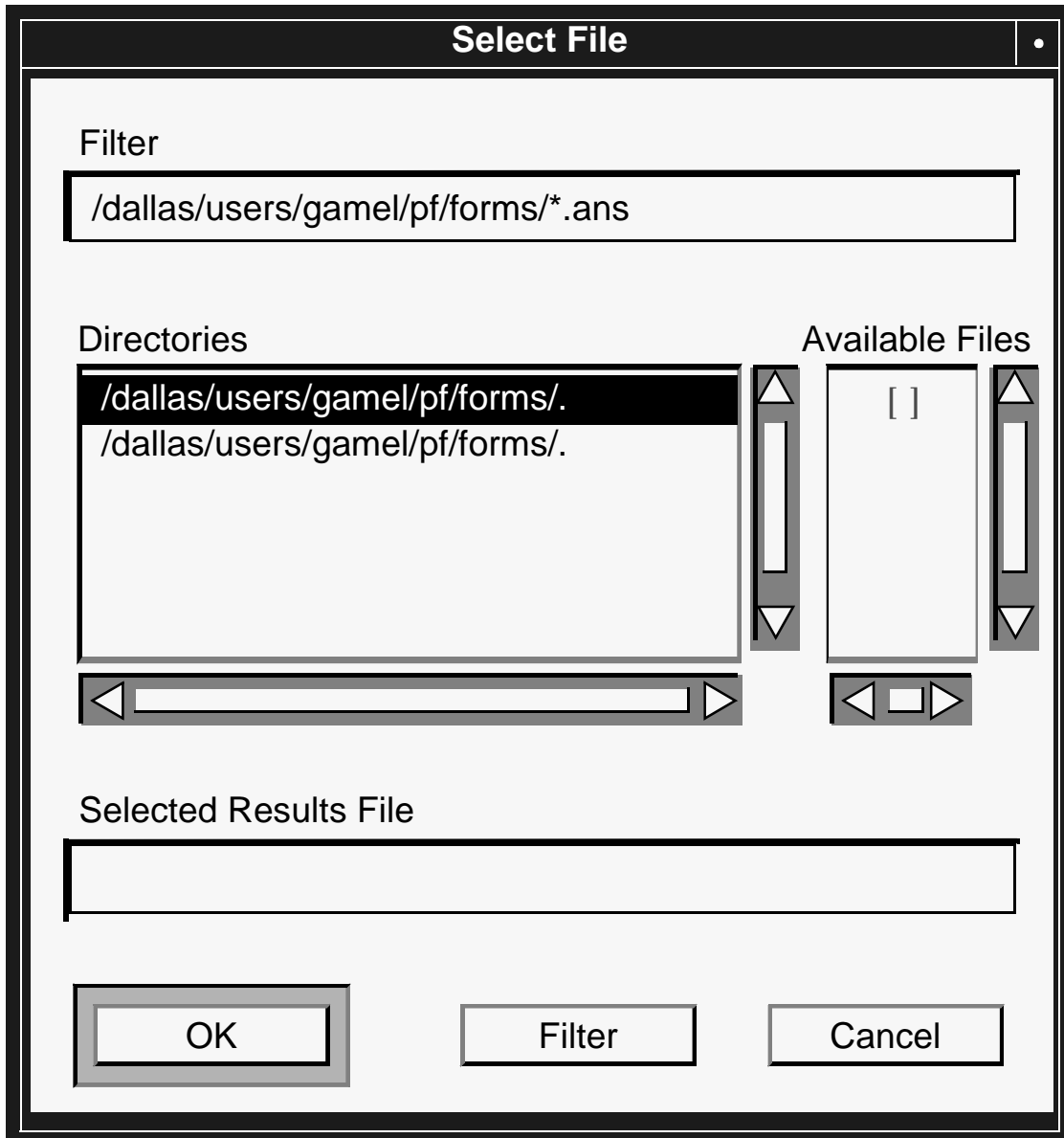


Click **OK**.

On the *Analysis* form, change the Action option menu to **Read Results File**.

Exercise 9

Click the **Select Results File ...** button and the following form should appear:



Sample Solution

```

/* $Header: /madrid/users9/pflib/pcl/custom/RCS/
           miracle_load_aom_data.pcl,v 1.2 93/02/21
           13:45:28 sprack Exp $ */

/

/*$sh */
/*
 * Purpose:
 * Define the option menu selections, button labels, and
 * button pcl classes for Miracle.
 */

CLASS miracle_load_aom_data

/*-----
*$ FUNCTION structural
*
* Purpose:
* Load the option menu data for "Miracle-Structural".
*/

FUNCTION structural(  num_actions, action_items,          @
                    num_objects, object_items,          @
                    num_methods, method_items,          @
                    num_buttons, button_labels,         @
                    button_callbacks, preference_class, @
                    callback_diagnostics                 )

/*
 * Local declarations:
 */
INTEGER num_actions,i
STRING action_items[]()
INTEGER num_objects()
STRING object_items[]()
INTEGER num_methods()
STRING method_items[]()
INTEGER num_buttons()
STRING button_labels[]()

```

Exercise 9

```
STRING button_callbacks[]()
STRING preference_class[]
LOGICAL callback_diagnostics
/*
 * Define the Actions, Objects and Methods. Note that at least
 * one actions must exist and for every action, one object
 * must exist. Methods are optional.
 */
num_actions = 2

action_items(1) = ***** 1 *****
action_items(2) = ***** 1 *****

num_objects(1) = ***** 1 *****
num_objects(2) = ***** 1 *****

object_items(1,1) = ***** 2 *****
object_items(2,1) = ***** 2 *****

num_methods(1,1) = ***** 2 *****
num_methods(2,1) = ***** 2 *****

method_items(1,1,1) = ***** 3 *****
method_items(2,1,1) = ***** 3 *****
/*
 * Define the number of buttons for each action-object
 * combinations. Then define the button labels and
 * callbacks.
 */
num_buttons(1,1) = ***** 3 *****
num_buttons(2,1) = ***** 3 *****
FOR( i = 1 TO ***** 3 *****
  button_labels(1,i,1) = ***** 3 *****
  button_labels(1,i,2) = ***** 3 *****
  button_callbacks(1,i,1) = ***** 3 *****
  button_callbacks(1,i,2) = ***** 3 *****
END FOR
```

```

button_labels(2,1,1) = "Select Results File..."
button_callbacks(2,1,1) = "analysis_select_file"

button_labels(2,1,1) = "Select Results File..."
button_callbacks(2,1,1) = "analysis_select_file"

/*
* Define the class for general button functions, such as
* the "apply" function.
*/
preference_class = ""

callback_diagnostics = FALSE

END FUNCTION /* structural */

END CLASS /* miracle_load_aom_data */

```

Solutions

```

1)action_items(1) = "Analyze"
   action_items(2) = "Read Results File"
   num_objects(1) = 1
   num_objects(2) = 1
2)object_items(1,1) = "Entire Model"
   object_items(2,1) = "Results Entities"
   num_methods(1,1) = 1
   num_methods(2,1) = 1
3)method_items(1,1,1) = "Full Run"
   method_items(2,1,1) = "Translate"
   num_buttons(1,1) = 2
   num_buttons(2,1) = 1
   FOR i = 1 TO 2
     button_labels(1,i,1) = "Solution Parameters..."
     button_labels(1,i,2) = "Select Load Cases..."
     button_callbacks(1,i,1) = "solution_param"
     button_callbacks(1,i,2) = "miracle_loadcases"

```