


EXERCISE 2



Writing Data for Nodes in a Group to a File

Objectives:

- Write a PCL function which writes node data to a file for all nodes in a group.



Problem Description:

In this Exercise we write a PCL function which writes the node id, reference CID, analysis CID and nodal coordinates to a file for all nodes in a group.

Suggested Exercise Steps:

- Write a PCL function using the online editor.
- Compile the function
- Verify the PCL function

Exercise Procedure:

Write a PCL Function

1. Either use vi or jot as the text editing tool. Open a file named *write_nodes_in_group.pcl*.

The function header follows:

```
FUNCTION write_nodes_in_group( group_id, filename )
/ *
* Write the following data for the nodes in a group out to a file
*   node id (I10), ref.cid (I10), Analysis CID (I10),
*   XYZ (F14.5)
* INPUT:
*   group_id    INTEGER  Group containing nodes of interest
*   filename    STRING   Filename to write the nodes to, the
*                           extension
*                           of .nod is added in this routine.
* Side Effects:
*   The specified file is created and filled with the nodes data
*/
(fill in)
END FUNCTION
```

Fill in the body of the function.

2. Compile the **write_nodes_in_group.pcl**.

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

After the main menu and command window appear, type
!!input write_nodes_in_group.pcl in the command
line:



Control Panel

Type here

Command Window

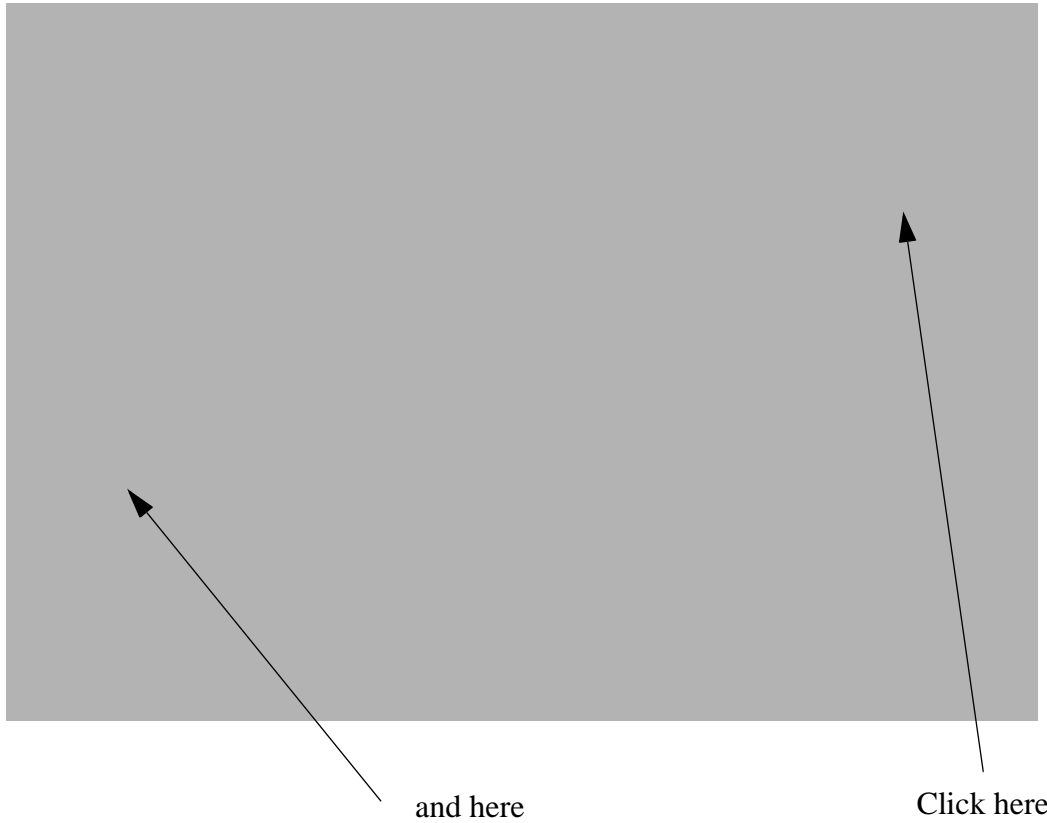


Resolve any compile errors by editing the PCL function
and re-compiling.

3. Test the function.

Exercise 2

Select **Play Session File...** from the *File menu*.



Wait until p3 finishes the session. You should see a patch appear in the viewport and then the patch is meshed with quad elements.

4. Verify the result:

In your local directory, a file named **exercise_2.nod** should have been created. Type:

>more exercise_2.nod

The following should appear in the xterm:
more exercise_2.nod

| node id | ref cid | Analysis | x coord | y coord | z coord |
|---------|---------|----------|---------|---------|---------|
| 16 | 0 | 0 | 1.00000 | 1.00000 | 0.00000 |
| 15 | 0 | 0 | 0.66667 | 1.00000 | 0.00000 |
| 14 | 0 | 0 | 0.33333 | 1.00000 | 0.00000 |
| 13 | 0 | 0 | 0.00000 | 1.00000 | 0.00000 |
| 12 | 0 | 0 | 1.00000 | 0.66667 | 0.00000 |
| 11 | 0 | 0 | 0.66667 | 0.66667 | 0.00000 |
| 10 | 0 | 0 | 0.33333 | 0.66667 | 0.00000 |
| 9 | 0 | 0 | 0.00000 | 0.66667 | 0.00000 |
| 8 | 0 | 0 | 1.00000 | 0.33333 | 0.00000 |
| 7 | 0 | 0 | 0.66667 | 0.33333 | 0.00000 |
| 6 | 0 | 0 | 0.33333 | 0.33333 | 0.00000 |
| 5 | 0 | 0 | 0.00000 | 0.33333 | 0.00000 |
| 4 | 0 | 0 | 1.00000 | 0.00000 | 0.00000 |
| 3 | 0 | 0 | 0.66667 | 0.00000 | 0.00000 |
| 2 | 0 | 0 | 0.33333 | 0.00000 | 0.00000 |
| 1 | 0 | 0 | 0.00000 | 0.00000 | 0.00000 |



Exercise 2



Sample Solution

```
FUNCTION write_nodes_in_group( group_id, filename )
/*
* Write all the nodes in a group out to a file
*
* INPUT:
* group_id INTEGER Group containing nodes of interest
* filename STRING Filename to write the nodes to, the extension
* of .nod is added in this routine.
* Side Effects:
* The specified file is created and filled with the nodes data
*/

STRING filename[]

INTEGER group_id, status, channel

INTEGER num_nodes, node_ids(VIRTUAL), acid(VIRTUAL)
INTEGER rcid(VIRTUAL), i, int_array(3)
REAL xyz(VIRTUAL)

status = text_open( filename//".nod", "NW", 0, 0, channel )

IF( status != 0 ) THEN
IF( ui_read_logical("File "//filename//".nod exists. "//@
    "Do you want to overwrite it?")) @
THEN
file_delete( filename//".nod" )
text_open( filename//".nod", "NW", 0, 0, channel )
ELSE
RETURN status
END IF
END IF

/*
* Count the number of nodes in the group
*/
```

Exercise 2

```
status = db_count_nodes_in_group( group_id, num_nodes )
IF( status != 0 ) THEN
msg_to_form( status, 4, appcode(status), 1, 1., "" )
RETURN status
END IF

/*
* Allocate memory to store the node ids
*/

sys_allocate_array( node_ids, 1, num_nodes )

/*
* Get all the nodes in a group
*/

status = db_get_all_node_ids_in_group( num_nodes, group_id, @
node_ids )
IF( status != 0 ) THEN
msg_to_form( status, 4, appcode(status), 1, 1., "" )
RETURN status
END IF

sys_allocate_array( rcid, 1, num_nodes )
sys_allocate_array( acid, 1, num_nodes )
sys_allocate_array( xyz, 1, num_nodes, 1, 3 )
status = db_get_nodes( num_nodes, node_ids, rcid, acid, xyz )
IF( status != 0 ) THEN
msg_to_form( status, 4, appcode(status), 1, 1., "" )
RETURN status
END IF

/*
* Write each record out
*/
```

```
text_write_string( channel, " node id ref cid "//      @
                  "Analysis x coord y coord"//      @
                  " z coord" )

FOR( i = 1 TO num_nodes )

int_array(1) = node_ids(i)
int_array(2) = rcid(i)
int_array(3) = acid(i)

status = text_write( channel, "%3I10% %3F14.5%",      @
                    int_array, xyz(i,1:3), ""      )
IF( status != 0 ) THEN RETURN status
END FOR

/*
* Close the external file
*/

text_close( channel, " " )

Write(str_from_integer( num_nodes )// " nodes written to file "// @
      filename // ".nod.")
RETURN 0

END FUNCTION
```