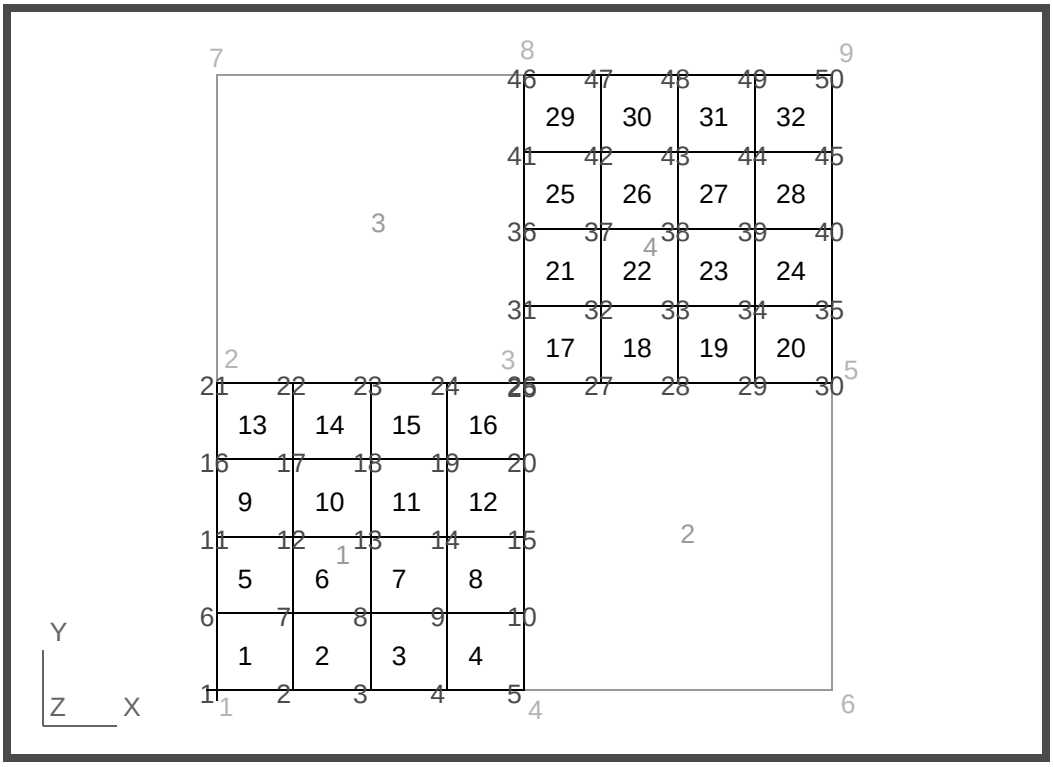
EXERCISE 11

List Processor



Objectives:

- Utilizes the List Processor to count the number of entities of a specific type and identify their label numbers from a list of entities.

Exercise Description:

In this exercise, you will create a function that uses the list processor to sort out entities of the type element and node from a list of entities. It will let you know how many elements or nodes exist in the the list and what their corresponding label numbers are. The function will print error messages if the list of entities contains invalid data.

Exercise Procedure:

1. Go to `ex11` directory. Enter the vi editor and edit the PCL function in a file called `exercise_11.template`. Substitute the appropriate PCL syntax or create new function for the lines which contain:

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

   After filling in the blanks, save the function and name the file `pick_labels.pcl`.

2. Compile the function. Run the C Preprocessor first:

/usr/lib/cpp -I/patran/patran3/customization pick_labels.pcl pick_labels.cpp

For the machine specific commands refer to the previous exercises

Start MSC/PATRAN by typing p3 in your xterm window.

Enter the command !!input pick_labels.cpp into the MSC/PATRAN command window.

All the error messages and diagnostics will be written to the MSC/PATRAN command window.

3. Run the session file called `surfaces.ses` to create surfaces and finite elements. Test the function yourself by typing in the command window:

   integer count
   integer ids(virtual)
   get_ids("elem 1t# node1t# surface 1t# point1t#", "elems", count, ids)
   dump count, ids

The command window will display the number of elements found and their label numbers.

Exercise Template:

```
#include "/patran/patran3/customization/lpenums.i"

FUNCTION get_ids( picklist, type, count, ids )

/*
 * Purpose: Given a picklist consisting of entities, identify the ones
 * that belong to the type specified and also their id numbers
 * In this example the types are node,elements or any
 *
 * Input:  A range of entities in picklist and the entity type in type
 *
 * Output: total number of entities found in count and their id number in
 * an array of ids
 *
 */
  STRING PICKLIST[], TYPE[]
  INTEGER IDS(VIRTUAL), COUNT
  INTEGER handle, status, list_count
  INTEGER lab(1), filter, list_type
/*
 * Some people consider it a bad style for a function to initialize an
 * output variable.  I chose to ignore that consideration in this example
 */

  count = 0

/*
 * Determine the filter to use depending on the type specified
 *
 */

  SWITCH ( type )
     CASE ("nodes", "NODE")
        /* Get the filter for nodes */
        filter = ***** 1 *****
     CASE ("ELEMENT", "elems", "elem", "elements")
        /* Get the filter for elements */
        filter = ***** 2 *****
     DEFAULT
        filter = LP_SUBLIST_ANY
  END SWITCH

  /*
   * Evaluate the picklist by label
   */

  status = lp_eval( ***** 3 ***** )
  IF ( status !=0 ) THEN
     ui_writec("Error in lp_eval\n")
     return status
  END IF

  /*
   * Count how many entities are of the specified type
   */

  status = lp_sublist_count(********** 4 **********)
  IF ( status != 0 ) THEN
     msg_to_form(status, 2, 0, 0, 0., "")
     RETURN status
  END IF
```

```
   status = sys_allocate_array(ids, 1, list_count)
   IF ( status != 0 ) THEN
      msg_to_form(status, 2, 0, 0, 0., "")
      RETURN status
   END IF

   WHILE ( count < list_count ) oi_loop
     /*
      * Check to make sure the sublist_type is equal to the type specified above
      */
      status = lp_sublist_type(handle, LP_SUBLIST_ANY, list_type )
      IF ( status != 0 ) THEN
         msg_to_form(status, 2, 0, 0, 0., "")
         RETURN status
      END IF
      IF ( list_type == filter ) THEN
         /*
          * Get the label number of the entity matching the type
          */
         status = lp_sublist_attribute_get_int(********** 5 ***********)
         IF ( status != 0 ) THEN
            status = lp_sublist_next(handle)
            IF ( status != 0 ) THEN
               IF ( status != 14000015 ) THEN
                  ui_writec("Error in evaluating lp_sublist_next()\n")
                  ui_writec("count = %d, lcount = %d, status = %d\n", @
                            count, list_count, status)
               END IF
               BREAK oi_loop
            END IF
            continue oi_loop
         END IF
         count += 1
         ids(count) = lab
      END IF


      /* Go to the next entity */
      status = lp_sublist_next(handle)
      IF ( status != 0 ) THEN
         IF ( status != 14000015 ) THEN
            ui_writec("Error in evaluating lp_sublist_next()\n")
            ui_writec("count = %d, lcount = %d, status = %d\n", @
                      count, list_count, status)
         END IF
         BREAK oi_loop
      END IF
   END WHILE

   IF ( list_count > count ) THEN
      sys_reallocate_array(ids, 1, count)
   END IF

   RETURN 0

END FUNCTION /* get_ids */
```

```
*1* LP_SUBLIST_NODE
*2* LP_SUBLIST_ELEMENT
*3* picklist, LP_EVAL_FOR_LABEL, handle
*4* handle, filter, list_count
*5* handle, LP_ATTRIBUTE_LABEL, lab
```