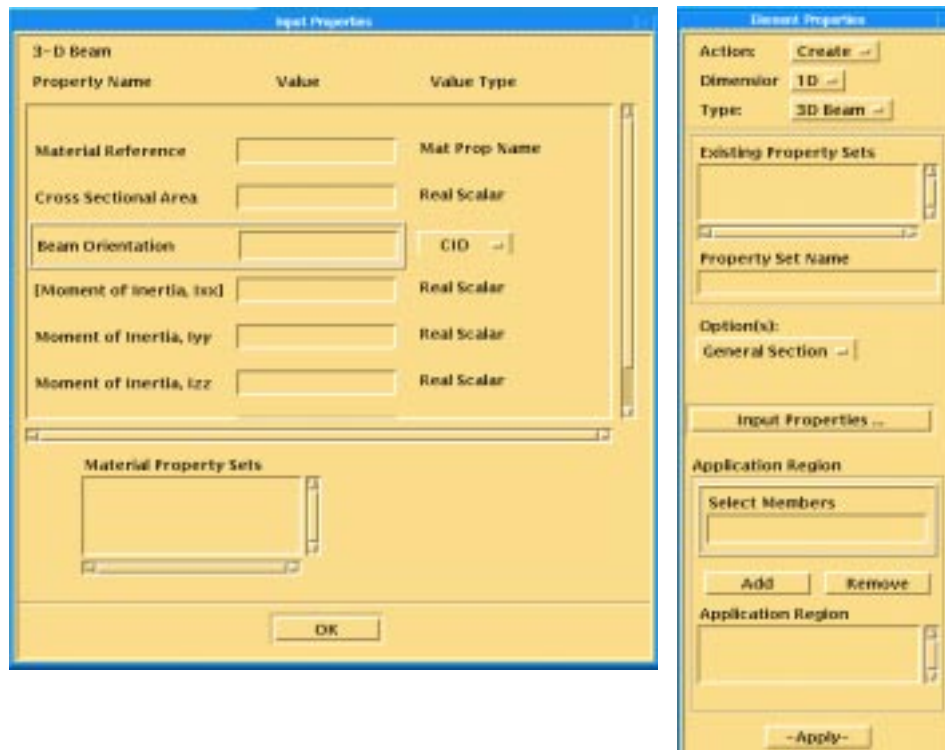


## EXERCISE 6

# *Loading Element Property Attributes*



### Objectives:

- Write a PCL function to load a new element property into p3 database.
- Use QLI to verify the new database relations and attributes created by adding the 3-D beam property set.



**Problem Description:**

In this Exercise, you will add a new element property set to an existing template PCL file named *exercise\_06.pcl*. In particular, you will add a 3-D beam with the following element property attributes.

Material Reference	(required data)
Cross sectional area	(required data)
Beam orientation	(required data)
Moment of Inertia, $I_{xx}$	(optional data)
Moment of Inertia, $I_{yy}$	(required data)
Moment of Inertia, $I_{zz}$	(required data)
Shear factor, $S_x$	(optional data)

Note that cross sectional area,  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  data boxes should not accept negative numbers, and shear factor data box should not accept any numbers greater than 1.0.

The screenshot shows the 'Input Properties' dialog box for a 3-D Beam element. The dialog has a title bar 'Input Properties' and a main area with a table of property names and value types. Below the table is a 'Material Property Sets' list and an 'OK' button.

Property Name	Value	Value Type
Material Reference	<input type="text"/>	Mat Prop Name
Cross Sectional Area	<input type="text"/>	Real Scalar
Beam Orientation	<input type="text"/>	CID <input type="text"/>
[Moment of Inertia, $I_{xx}$ ]	<input type="text"/>	Real Scalar
Moment of Inertia, $I_{yy}$	<input type="text"/>	Real Scalar
Moment of Inertia, $I_{zz}$	<input type="text"/>	Real Scalar

Material Property Sets

OK

---

## Suggested Exercise Steps:

- Edit *exercise\_06.template* file and make modifications to add the new element property attributes.
- After replacing the “template” suffix to *.pcl*, compile the PCL program. Be sure to use the C pre-processor.
- Open a new database using the *base.db* template.
- Run **exercise\_06.pcl** and quit *p3*
- Verify the form.
- Verify the database relations/attributes.

## Exercise Procedure:

1. Either use *vi* or *jot* as the text editing tool. Open the file named *exercise\_06.template*. It should already exist in your directory. Make the appropriate changes to add the 3-D beam element property set, and change the file name to **exercise\_06.pcl**.
2. Compile the function.

Type in the command line:

```
% /usr/lib/cpp -P exercise_06.pcl exercise_06.cpp  
% p3
```

Type *p3* in your *xterm*. At the command line type:

```
!!input exercise_06.cpp  
Resolve any compiler errors.
```

In the *Control Panel*, select **File** from the *top menu bar*.  
Select **New Database...** from the pull down menu.

**New Database**

Template Database Name  
/patran/patran3/template.db

Change Template ...

Modify Preferences...

Filter  
/training\_docs/ivory/pat304/\*.db

Directories  
/pat304/.  
/pat304/..

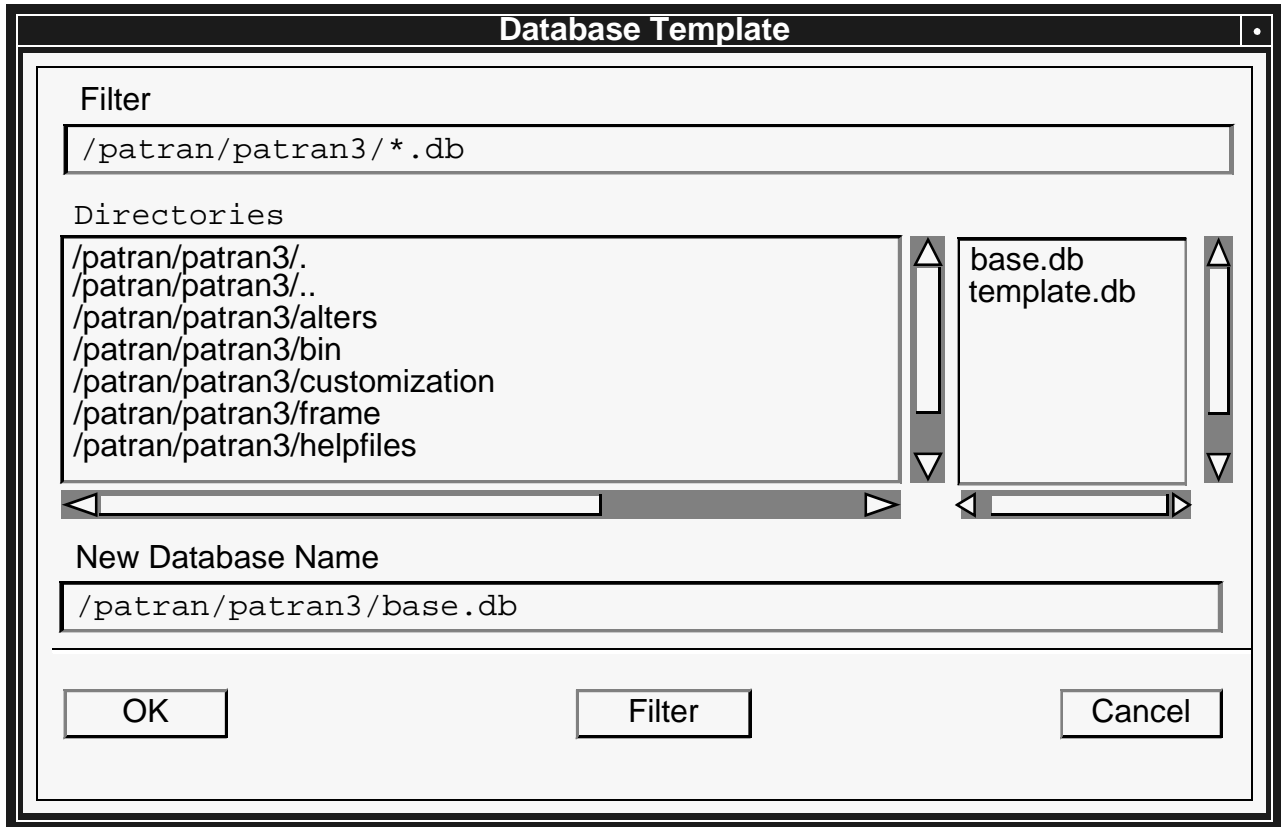
[ ]

New Database Name

OK Filter Cancel

Click the **Change Template...** button.

In the *Database Template* form, select **base.db** from the Database Template List listbox.



Click the **OK** button.

In the *New Database* form, click in the New Database Name databox and enter the name **miracle\_template.db**.

**New Database**

Template Database Name  
/patran/patran3/base.db

Change Template ...

Modify Preferences...

Filter  
/training\_docs/ivory/pat304/\*.db

Directories

/pat304/.  
/pat304/..

[ ]

New Database Name  
miracle\_template

OK Filter Cancel

Click the **OK** button.

Now type **load\_miracle()** in the command line and return.

Close the database.

Quit p3.

3. Test your modifications to exercise\_06.pcl

Enter p3 at the prompt.

Click on the **File** in the *top menu bar*. Select **New Database...** from the pull-down menu. In the *New Database* form, change the Template Database Name to **miracle\_template.db**. Change the New Database Name to **exercise\_6.db**.

The image shows a 'New Database' dialog box with the following fields and controls:

- Template Database Name:** A text box containing 'miracle\_template.db' and a 'Change Template ...' button below it.
- Modify Preferences...**
- Filter:** A text box containing '/training\_docs/ivory/pat304/\*.res'.
- Directories:** A list box containing '/pat304/' and '/pat304/..'.
- New Database Name:** A text box containing 'exercise\_6.db'.
- Buttons:** 'OK', 'Filter', and 'Cancel' buttons at the bottom.

Click the **OK** button.

4. Verify the result.

Select the **Element Props** radio button in the *Control Panel*.

Set the *Action, Dimension, Type* option menus to **Create, 1D, 3D Beam**, respectively.

---

The form should appear as follows:

The image shows a screenshot of the 'Element Properties' dialog box. At the top, there are three dropdown menus: 'Action:' set to 'Create', 'Dimensio' set to '1D', and 'Type:' set to '3D Beam'. Below these is a section titled 'Existing Property Sets' which contains a list box and a 'Property Set Name' text field. In the middle of the dialog is a button labeled 'Input Properties ...'. Below this is an 'Application Region' section containing a 'Select Members' list box, 'Add' and 'Remove' buttons, and another 'Application Region' text field. At the bottom of the dialog is an '-Apply-' button. An arrow points from the text 'Click here' to the 'Input Properties ...' button.

Click the **Input Properties ...** button.

The following form should appear:



---

Now, go back to the Element Properties form and change the Dimension option menu to **2D**. Once again, click the **Input Properties ...** button and you should see:



Finally, set the Dimension to **3D** and click the **Input Properties ...** button. The *Input Properties* form should appear as follows:



5. Using QLI, inspect the entries you have just created in the database. The following are some of the relations (i.e. Tables) you may want to examine.

- Selected\_element\_Type
- Selected\_property\_set
- Allowable\_phys\_prop
- Phys\_prop\_data
- Phys\_prop\_set\_mbr\_defn
- Phys\_prop\_set\_defn

To list all of the relations in the p3 database, use the following command

QLI> show relations

We have included a sample QLI session for your reference. However, feel free to let your curiosity take you to unfamiliar relations!

```
dresden_%qli
Welcome to QLI
Query Language Interpreter
QLI> ready miracle_template.db
QLI> show relation selected_element_type
SELECTED_ELEMENT_TYPE
  ANALYSIS_TYPE_ID          short binary
  ANALYSIS_CODE_ID         short binary
  GEOMETRIC_OPTION_CODE     short binary
  CONDENSE_OPTION_CODE     short binary
  FORMULATION_OPTION_CODE  short binary
  LAMINATE_OPTION_CODE     short binary
  ELEMENT_TOPOLOGY_ID     long binary
  DOF_SET_ID               short binary
  MATERIAL_LINEARITY_CODE  short binary
  MATERIAL_DIRECTIONALITY_CODE short binary
  ELEMENT_TYPE_ID         long binary
  CONFIG_NTL_25           short binary
Triggers defined for this relation:
  INSERT_PRO_SEL_ELEM_TYPE  Pre-store, Sequence 0, Active
  UPDATE_PRO_SEL_ELEM_TYPE  Pre-modify, Sequence 0, Active
QLI> print SELECTED_ELEMENT_TYPE

ANALYSIS ANALYSIS GEOMETRIC CONDENSE FORMULATION LAMINATE ELEMENT DOF
TYPE      CODE    OPTION  OPTION  OPTION    OPTION  TOPOLOGY SET
ID        ID      CODE   CODE   CODE     CODE   ID       ID
-----
      1      1      2      42      1      2      2      20
1        1      36      1
1        1      25      24      1      1      4      19
1        1      54      1
1        1      25      30      1      1      28     15
1        1      71      0

QLI> print allowable_phys_prop
```

**EXERCISE 6**

# *Loading Element Property Attributes*

---

ANALYSIS CODE ID	PHYS PROP ID	PHYS PROP ALIAS	REQUIRED FLAG	NEUTRAL FILE ORDER	DEFAULT DATA TYPE
1	13	Material Reference	1	0	5
1	1	Cross Sectional Area	1	0	1
prop > 0.					
1	36	Plate Thickness	1	0	1
prop > 0.					
1	2	Beam Orientation	1	0	9
1	17	Moment of Inertia, Ixx	0	0	1
prop > 0.					
1	18	Moment of Inertia, Iyy	0	0	1
prop > 0.					
1	19	Moment of Inertia, Izz	0	0	1
prop > 0.					
1	1004	Shear Factor, Sx	1	0	1

prop < 1.

QLI> print phys\_prop\_set\_mbr\_defn where phys\_prop\_set\_defn\_id = 10002

PHYS PROP SET DEFN ID	PHYS PROP ID	NEUTRAL FILE ORDER
10002	13	1
10002	1	2
10002	2	3
10002	17	4
10002	18	5
10002	19	6
10002	1004	7

## Sample Solution

```

#include "app_ep_defn_ids.i"
#include "app_ep_prop_ids.i"
#include "mat_ind_cons.i"
#include "mat_words.p"
#define _YES 0
#define _NO 1
#define _NOT_VALID 0
#define _SCALAR 1
#define _VECTOR 2
#define _INTEGER 3
#define _STRING 4
#define _MATERIAL 5
#define _LIST 6
#define _FIELD 7
#define _NODAL 8
#define _COORD 9
#define _STRUCTURAL 1
#define _THERMAL 2

FUNCTION load_miracle()

    INTEGER status, bar_code, plate_code, solid_code, anal_code_id

    /*
     * Fetch the first unused analysis code id
     */

    status = analysis_main.get_id_for_new_code ( anal_code_id )
    IF( status != 0 ) THEN
        write("Call to analysis_main.get_id_for_new_code failed.")
        msg_to_form( status, 4, appcode(status), 1, 1., "" )
        RETURN
    END IF
    /*
     * Define new analysis code
     */

    status = db_create_analysis_code( anal_code_id, "Miracle", ".inp", ".ans", 2, @
                                     [_STRUCTURAL,_THERMAL], _STRUCTURAL )
    IF( status != 0 ) THEN
        write("Call to db_create_analysis_code failed.")
        msg_to_form( status, 4, appcode(status), 1, 1., "" )
        RETURN
    END IF

    /*
     * Define the generic element types, element properties and material properties
     */

    load_generics()

    /*
     * Get P3/PATRAN topology codes
     */
    status = fem_get_patran25_etop(***** 1 *****)
    IF( status != 0 ) THEN
        write("Call to fem_get_patran25_etop failed.")
        msg_to_form( status, 4, appcode(status), 1, 1., "" )
        RETURN
    END IF

    status = fem_get_patran25_etop(4,4,plate_code)
    IF( status != 0 ) THEN
        write("Call to fem_get_patran25_etop failed.")
        msg_to_form( status, 4, appcode(status), 1, 1., "" )
        RETURN
    END IF

```

```

status = fem_get_patran25_etop(8,8,solid_code)
IF( status != 0 ) THEN
  write("Call to fem_get_patran25_etop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "")
  RETURN
END IF

/*
 * Define element types, begin with a beam
 */

status = db_create_selected_etype_wc ( ***** 2 ***** @ /* Analysis type id */
***** 2 ***** @ /* Analysis code */
***** 2 ***** @ /* element geometric option */
***** 2 ***** @ /* Element condensation option */
***** 2 ***** @ /* Element formulation */
***** 2 ***** @ /* laminate option */
***** 2 ***** @ /* topology code */
***** 2 ***** ,@ /* degree-of-freedom set */
***** 2 ***** @ /* material linearity code */
***** 2 ***** @ /* material directionality code */
***** 2 ***** @ /* generic element type */
***** 2 ***** /* patran 2.5 config. code */

IF( status != 0 ) THEN
  write("Call to db_create_selected_etype_wc failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "")
  RETURN
END IF
status = db_create_selected_etype_wc ( _STRUCTURAL, @ /* Analysis type id */
anal_code_id, @ /* Analysis code */
ID_GEO_N_A, @ /* element geometric option */
ID_COND_PLATE, @ /* Element condensation option */
ID_FORM_STANDARD, @ /* Element formulation */
ID_HOMOGENEOUS, @ /* laminate option */
plate_code, @ /* topology code */
ID_DOF_UXUYUZRXY, @ /* degree-of-freedom set */
MAT_LIN_ELASTIC, @ /* material linearity code */
MAT_DIR_ISOTROPIC, @ /* material directionality code */
ID_NAME_PLATE, @ /* generic element type */
1 ) /* patran 2.5 config. code */

IF( status != 0 ) THEN
  write("Call to db_create_selected_etype_wc failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "")
  RETURN
END IF

/*
 * define a solid hex
 */

status = db_create_selected_etype_wc ( _STRUCTURAL,@ /* Analysis type id */
anal_code_id, @ /* Analysis code */
ID_GEO_N_A, @ /* element geometric option */
ID_COND_SOLID, @ /* Element condensation option */
ID_FORM_STANDARD, @ /* Element formulation */
ID_HOMOGENEOUS, @ /* laminate option */
solid_code, @ /* topology code */
ID_DOF_UXUYUZ, @ /* degree-of-freedom set */
MAT_LIN_ELASTIC, @ /* material linearity code */
MAT_DIR_ISOTROPIC, @ /* material directionality code*/
ID_NAME_SOLID, @ /* generic element type */
0 ) /* patran 2.5 config. code */

IF( status != 0 ) THEN
  write("Call to db_create_selected_etype_wc failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "")
  RETURN

```

```

END IF

/*
 * Re-create element type summary table
 */

status = ***** 3 *****
IF( status != 0 ) THEN
  write("Call to elementprops_def_create.make_lem_summary failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * define a plate
 */
/*
 * Define element property words
 */

status = db_create_allowable_phys_prop( ***** 4 ***** @ /* Analysis code id */
                                     ID_PROP_MATERIAL_NAME, @ /* property definition word */
                                     ***** 4 ***** , @ /* Name to be displayed */
                                     ***** 4 ***** @ /* Optional? */
                                     ***** 4 ***** @ /* Allowable datatype */
                                     ***** 4 ***** @ /* Default type */
                                     ***** 4 ***** @ /* Location in 2.5 PFEG record*/
                                     ***** 4 ***** @ /* PCL range specifier (INT) */
                                     ***** 4 ***** @ /* PCL range specifier (REAL) */
                                     ***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("1st Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id */
                                     ***** 4 ***** @ /* property definition word */
                                     "Cross Sectional Area",@ /* Name to be displayed */
                                     ***** 4 ***** @ /* Optional? */
                                     ***** 4 ***** @ /* Allowable datatype */
                                     ***** 4 ***** @ /* Default type */
                                     ***** 4 ***** @ /* Location in 2.5 PFEG record*/
                                     ***** 4 ***** @ /* PCL range specifier (INT) */
                                     ***** 4 ***** @ /* PCL range specifier (REAL) */
                                     ***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("2nd Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id */
                                     ***** 4 ***** @ /* property definition word */
                                     "Plate Thickness", @ /* Name to be displayed */
                                     ***** 4 ***** @ /* Optional? */
                                     ***** 4 ***** @ /* Allowable datatype */
                                     ***** 4 ***** @ /* Default type */
                                     ***** 4 ***** @ /* Location in 2.5 PFEG record*/
                                     ***** 4 ***** @ /* PCL range specifier (INT) */
                                     ***** 4 ***** @ /* PCL range specifier (REAL) */
                                     ***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("3rd Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

```

```

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id      */
***** 4 ***** @ /* property definition word */
"Beam Orientation", @ /* Name to be displayed */
***** 4 ***** @ /* Optional? */
***** 4 ***** @ /* Allowable datatype */
***** 4 ***** @ /* Default type */
***** 4 ***** @ /* Location in 2.5 PFEG record*/
***** 4 ***** @ /* PCL range specifier (INT) */
***** 4 ***** @ /* PCL range specifier (REAL) */
***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("4th Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id      */
***** 4 *****X, @ /* property definition word */
"Moment of Inertia, Ixx",@ /* Name to be displayed */
***** 4 ***** @ /* Optional? */
***** 4 ***** @ /* Allowable datatype */
***** 4 ***** @ /* Default type */
***** 4 ***** @ /* Location in 2.5 PFEG record */
***** 4 ***** @ /* PCL range specifier (INT) */
***** 4 ***** @ /* PCL range specifier (REAL) */
***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("5th Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id      */
***** 4 *****YY, @ /* property definition word */
"Moment of Inertia, Iyy",@ /* Name to be displayed */
***** 4 ***** @ /* Optional? */
***** 4 ***** @ /* Allowable datatype */
***** 4 ***** @ /* Default type */
***** 4 ***** @ /* Location in 2.5 PFEG record */
***** 4 ***** @ /* PCL range specifier (INT) */
***** 4 ***** @ /* PCL range specifier (REAL) */
***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("6th Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id      */
***** 4 *****ZZ, @ /* property definition word */
"Moment of Inertia, Izz", @ /* Name to be displayed */
***** 4 ***** @ /* Optional? */
***** 4 ***** @ /* Allowable datatype */
***** 4 ***** @ /* Default type */
***** 4 ***** @ /* Location in 2.5 PFEG record */
***** 4 ***** @ /* PCL range specifier (INT) */
***** 4 ***** @ /* PCL range specifier (REAL) */
***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("7th Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

```

```

status = db_create_allowable_phys_prop( ***** 4 *****@ /* Analysis code id      */
                                     ***** 4 ***** , @ /* property definition word  */
                                     ***** 4 ***** , @ /* Name to be displayed      */
                                     ***** 4 ***** , @ /* Optional?                  */
                                     ***** 4 ***** , @ /* Allowable datatype        */
                                     ***** 4 ***** , @ /* Default type              */
                                     ***** 4 ***** , @ /* Location in 2.5 PFEG record */
                                     ***** 4 ***** , @ /* PCL range specifier (INT)  */
                                     ***** 4 ***** , @ /* PCL range specifier (REAL) */
                                     ***** 4 ***** ) /* PCL range specifier (STRING)*/

IF( status != 0 ) THEN
  write("8th Call to db_create_allowable_phys_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF
/*
 * Define property sets
 */

status = db_create_pp_set_defn_wnord ( 10001, @ /* Property set id */
                                     "Homogeneous Plate", @ /* Property set name */
                                     2, @ /* Number of valid props in set */
                                     [ID_PROP_MATERIAL_NAME, ID_PROP_THICKNESS ], @ /* List of valid props in set */
                                     [1,2] ) /* Neutral file order */

IF( status != 0 ) THEN
  write("Call to db_create_pp_set_defn_wnord failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_pp_set_defn_wnord ( ***** 5 *****@ /* Valid Property set id */
                                     "3-D Beam", @ /* Valid Property set name */
                                     ***** 5 ***** @ /* Number of valid props in set */
                                     ***** 5 ***** @
                                     ***** 5 *****N, @
                                     ***** 5 ***** @
                                     ***** 5 ***** @
                                     ***** 5 ***** @
                                     ***** 5 ***** @
                                     ***** 5 ***** @ /* List of valid props in set */
                                     ***** 5 ***** ) /* Neutral file order */

IF( status != 0 ) THEN
  write("Call to db_create_pp_set_defn_wnord failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_pp_set_defn_wnord ( 10003, @ /* Valid Property set id */
                                     "Homogeneous Solid", @ /* Valid Property set name */
                                     1, @ /* Number of valid props in set */
                                     [ID_PROP_MATERIAL_NAME], @ /* List of valid props in set */
                                     [1] ) /* Neutral file order */
                                     ***** 5 ***** @

IF( status != 0 ) THEN
  write("Call to db_create_pp_set_defn_wnord failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Associate property sets with element types
 */

status = db_create_selected_prop_set ( ***** 6 *****@ /* Analysis type      */
                                     ***** 6 ***** @ /* Analysis code      */
                                     ***** 6 ***** @ /* element geometric option */
                                     ***** 6 ***** @ /* Element condensation option */
                                     ***** 6 ***** @ /* Element formulation    */
                                     ***** 6 ***** @ /* laminate option       */
                                     ***** 6 ***** @ /* topology code        */
                                     ***** 6 ***** , @ /* degree-of-freedom set  */
                                     ***** 6 ***** ) /* Valid Property set id      */

```

```

IF( status != 0 ) THEN
  write("Call to db_create_selected_prop_set failed.")

  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_selected_prop_set ( _STRUCTURAL, @ /* Analysis type          */
                                     anal_code_id,   @ /* Analysis code          */
                                     ID_GEO_N_A,     @ /* element geometric option */
                                     ID_COND_PLATE,   @ /* Element condensation option */
                                     ID_FORM_STANDARD, @ /* Element formulation      */
                                     ID_HOMOGENEOUS, @ /* laminate option         */
                                     plate_code,     @ /* topology code           */
                                     ID_DOF_UXUYUZRXY, @ /* degree-of-freedom set   */
                                     10001           ) /* Valid Property set id   */

IF( status != 0 ) THEN
  write("Call to db_create_selected_prop_set failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_selected_prop_set ( _STRUCTURAL, @ /* Analysis type          */
                                     anal_code_id,   @ /* Analysis code          */
                                     ID_GEO_N_A,     @ /* element geometric option */
                                     ID_COND_SOLID,   @ /* Element condensation option */
                                     ID_FORM_STANDARD, @ /* Element formulation      */
                                     ID_HOMOGENEOUS, @ /* laminate option         */
                                     solid_code,     @ /* topology code           */
                                     ID_DOF_UXUYUZ,   @ /* degree-of-freedom set   */
                                     10003           ) /* Valid Property set id   */

IF( status != 0 ) THEN
  write("Call to db_create_selected_prop_set failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Define material categories
 */

status = db_create_matl_category ( anal_code_id, @ /* Analysis code          */
                                  MAT_CAT_ISOTROPIC, @ /* Material Category      */
                                  "Isotropic", @ /* Category name          */
                                  MAT_DIR_ISOTROPIC ) /* Material Directionality */

IF( status != 0 ) THEN
  write("Call to db_create_matl_category failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Make the material category valid
 */

status = db_create_valid_mtl_categories ( anal_code_id, @ /*Analysis code          */
                                          _STRUCTURAL, @ /* Analysis type          */
                                          [MAT_CAT_ISOTROPIC], @ /*List of material categories */
                                          1 ) /*Number of material categories */

IF( status != 0 ) THEN
  write("Call to db_create_valid_mtl_categories failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Define material models
 */

```

```

status = db_create_constitutive_model ( anal_code_id,@ /* Analysis code          */
                                       1,                @ /* Material model id          */
                                       "Linear Elastic",   @ /* Label                      */
                                       MAT_LIN_ELASTIC    ) /* Linearity code            */
IF( status != 0 ) THEN
  write("Call to db_create_constitutive_model failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Make the constitutive model valid
 */

status = db_create_valid_const_models ( anal_code_id,@ /* Analysis code          */
                                       _STRUCTURAL,     @ /* Analysis type          */
                                       MAT_CAT_ISOTROPIC, @ /* Material Category id  */
                                       [1],             @ /* List of material models */
                                       1                 ) /* Number of material models */
IF( status != 0 ) THEN
  write("Call to db_create_valid_const_models failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Define material words
 */

status = db_create_matl_prop_alias ( anal_code_id,@ /* Analysis code          */
                                     _STRUCTURAL,     @ /* Analysis type          */
                                     1,                @ /* Material category      */
                                     1,                @ /* Material Constitutive model */
                                     _REF_TEMP,        @ /* Material Word          */
                                     "Reference Temperature", @ /* Label (Alias)         */
                                     _NO_FIELD         ) /* Type of field          */
IF( status != 0 ) THEN
  write("Call to db_create_matl_prop_alias failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_matl_prop_alias ( anal_code_id, @ /* Analysis code          */
                                     _STRUCTURAL,     @ /* Analysis type          */
                                     1,                @ /* Material category      */
                                     1,                @ /* Material Constitutive model */
                                     _ELASTIC_MODULUS, @ /* Material Word          */
                                     "Young's Modulus", @ /* Label (Alias)         */
                                     _TEMP_DEP_FIELD   ) /* Type of field          */
IF( status != 0 ) THEN
  write("Call to db_create_matl_prop_alias failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_matl_prop_alias ( anal_code_id, @ /* Analysis code          */
                                     _STRUCTURAL,     @ /* Analysis type          */
                                     1,                @ /* Material category      */
                                     1,                @ /* Material Constitutive model */
                                     _POISSONS_RATIO,  @ /* Material Word          */
                                     "Poisson's Ratio", @ /* Label (Alias)         */
                                     _TEMP_DEP_FIELD   ) /* Type of field          */
IF( status != 0 ) THEN
  write("Call to db_create_matl_prop_alias failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

```

```

status = db_create_matl_prop_alias ( anal_code_id,      @ /* Analysis code      */
                                   _STRUCTURAL,        @ /* Analysis type      */
                                   1,                  @ /* Material category   */
                                   1,                  @ /* Material Constitutive model */
                                   _DENSITY,           @ /* Material Word       */
                                   "Mass Density",     @ /* Label (Alias)       */
                                   _TEMP_DEP_FIELD )    /* Type of field       */

IF( status != 0 ) THEN
  write("Call to db_create_matl_prop_alias failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_matl_prop_alias ( anal_code_id,@ /* Analysis code      */
                                   _STRUCTURAL,        @ /* Analysis type      */
                                   1,                  @ /* Material category   */
                                   1,                  @ /* Material Constitutive model */
                                   _THERMAL_EXPANSION, @ /* Material Word       */
                                   "Thermal Expansion Coeff.",@ /* Label (Alias)       */
                                   _TEMP_DEP_FIELD )    /* Type of field       */

IF( status != 0 ) THEN
  write("Call to db_create_matl_prop_alias failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

status = db_create_matl_prop_alias ( anal_code_id,@ /* Analysis code      */
                                   _STRUCTURAL,        @ /* Analysis type      */
                                   1,                  @ /* Material category   */
                                   1,                  @ /* Material Constitutive model */
                                   _STRUCTURAL_DAMPING, @ /* Material Word       */
                                   "Structural Damping Coeff.", @ /* Label (Alias)       */
                                   TEMP_DEP_FIELD )    /* Type of field       */

IF( status != 0 ) THEN
  write("Call to db_create_matl_prop_alias failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

/*
 * Define the material records
 */

status = db_create_allowable_matl_prop ( anal_code_id,@ /* Analysis code      */
/*
                                   _STRUCTURAL,        @ /* Analysis type      */
                                   1,                  @ /* Material Category   */
                                   1,                  @ /* Material Constitutive model */
                                   [0,0,0,0,0],        @ /* List of 5 option categories */
                                   [_REF_TEMP,         @
                                   _ELASTIC_MODULUS,    @
                                   _POISSONS_RATIO,    @
                                   _DENSITY,           @
                                   _THERMAL_EXPANSION, @
                                   _STRUCTURAL_DAMPING], @ /* List of material words      */
                                   6                    /* Number of material words    */

IF( status != 0 ) THEN
  write("Call to db_create_allowable_matl_prop failed.")
  msg_to_form( status, 4, appcode(status), 1, 1., "" )
  RETURN
END IF

$ Set the new Miracle to be the Default Analysis Code
uil_pref_analysis.set_analysis_pref( "Miracle", "Structural", ".inp", ".ans" )
$ End
END FUNCTION /* load_miracle */

```



## Solutions

```

1) status = fem_get_patran25_etop(2,2,bar_code)

2) status = db_create_selected_type_wc('STRUCTURAL',@ /* Analysis type id
anal_code_id,@ /* Analysis code
ID_PROP_MATERIAL_NAME,@ /* property definition word
'Material Reference',@ /* Name to be displayed
NO,@ /* Optional?
[MATERIAL,0,0,0,0,0,0,0,0],@ /* Allowable datatype
MATERIAL,@ /* Default type
Location in 2.5 PRG record
,@ /*
PCL range specifier (INT)
,@ /*
PCL range specifier (REAL)
,@ /*
PCL range specifier (STRING)
)
3) status = elementprops_def_create.make_elem_summary()

4) status = db_create_allowed_phys_props(anal_code_id,@ /* Analysis code id
ID_PROP_MATERIAL_NAME,@ /* property definition word
'Material Reference',@ /* Name to be displayed
NO,@ /* Optional?
[SCALAR,0,0,0,0,0,0,0,0],@ /* Allowable datatype
SCALAR,@ /* Default type
Location in 2.5 PRG record
,@ /*
PCL range specifier (INT)
,@ /*
PCL range specifier (REAL)
,@ /*
PCL range specifier (STRING)
)
status = db_create_allowed_phys_props(anal_code_id,@ /* Analysis code id
ID_PROP_THICKNESS,@ /* property definition word
'Plate Thickness',@ /* Name to be displayed
NO,@ /* Optional?
[SCALAR,0,0,0,0,0,0,0,0],@ /* Allowable datatype
SCALAR,@ /* Default type
Location in 2.5 PRG record
,@ /*
PCL range specifier (INT)
,@ /*
PCL range specifier (REAL)
,@ /*
PCL range specifier (STRING)
)
status = db_create_allowed_phys_props(anal_code_id,@ /* Analysis code id
ID_PROP_BEAM_ORIENTATION,@ /* property definition word
'Beam Orientation',@ /* Name to be displayed
NO,@ /* Optional?
[NO,COORD,0,0,0,0,0,0,0],@ /* Allowable datatype
COORD,@ /* Default type
Location in 2.5 PRG record
,@ /*
PCL range specifier (INT)
,@ /*
PCL range specifier (REAL)
,@ /*
PCL range specifier (STRING)
)
status = db_create_allowed_phys_props(anal_code_id,@ /* Analysis code id
ID_PROP_MOM_INERTIA_XX,@ /* property definition word
'Moment of Inertia_Ixx',@ /* Name to be displayed
YES,@ /* Optional?
[SCALAR,0,0,0,0,0,0,0,0],@ /* Allowable datatype
SCALAR,@ /* Default type
Location in 2.5 PRG record
,@ /*
PCL range specifier (INT)
,@ /*
PCL range specifier (REAL)
,@ /*
PCL range specifier (STRING)
)
status = db_create_allowed_phys_props(anal_code_id,@ /* Analysis code id
ID_PROP_MOM_INERTIA_YX,@ /* property definition word
'Moment of Inertia_Ixy',@ /* Name to be displayed
YES,@ /* Optional?
[SCALAR,0,0,0,0,0,0,0,0],@ /* Allowable datatype
SCALAR,@ /* Default type
Location in 2.5 PRG record
,@ /*
PCL range specifier (INT)
,@ /*
PCL range specifier (REAL)
,@ /*
PCL range specifier (STRING)
)

```

```

status = db.create_allowed_phys_prop( anal_code_id,@ /* Analysis code id
/* property definition word
/* Name to be displayed
/* Optional?
/* Allowable datatype
/* Default type
/* Location in 2.5 PREG record
/* PCL range specifier (INT)
/* PCL range specifier (REAL)
/* PCL range specifier (STRING)
)
status = db.create_allowed_phys_prop( anal_code_id,@ /* Analysis code id
/* property definition word
/* Name to be displayed
/* Optional?
/* Allowable datatype
/* Default type
/* Location in 2.5 PREG record
/* PCL range specifier (INT)
/* PCL range specifier (REAL)
/* PCL range specifier (STRING)
)
5)status = db.create_pp_set_defn_wword ( 10002,
@ /* Valid property set name *
@ /* Number of valid props in set *
ID_PROP_MATERIAL_NAME,
ID_PROP_AREA,
ID_PROP_BEAM_ORIENTATION,
ID_PROP_MOM_INERTIA_XX,
ID_PROP_MOM_INERTIA_YY,
ID_PROP_MOM_INERTIA_ZZ,
ID_PROP_SHEAR_FACTOR ],
[1,2,3,4,5,6,7] )
/* Neutral file order *
6)status = db.create_selected_prop_set ( _STRUCTURAL, @ /* Analysis type
anal_code_id,
@ /* Analysis code
ID_GEO_GEN_SECT,
@ /* element geometric option
ID_COND_3D_BEAM,
@ /* element condensation option
ID_FORM_STANDARD,
@ /* Element formulation
ID_LAM_N_A,
@ /* laminate option
bar_code,
@ /* topology code
ID_DOF_UXUYUZRXRYRZ,
@ /* degree-of-freedom set
10002)
/* Valid property set id

```