



Patran

Grafische Basisobjekte in den Patran Viewport einfügen

Grafische Basisobjekte

Eine Standardfunktionalität ist das Erstellen von Grafischen Basisobjekten auf dem Patran Viewport.

Anders als bei Titeln bewegen sich die Grafikobjekte mit dem Modell mit. Ein im Elementschwerpunkt platzierter Text dreht sich dementsprechend mit. Einem Grafischen Segment können verschiedene Graphische Objekte zugeordnet werden. Dadurch wird eine bessere Performance gewährleistet, als würde ich jeweils ein eigenes Grafisches Segment erzeugen.

Mit Hilfe Graphischer Basisobjekte können Texte, Symbole oder Linien dargestellt werden. Mehr Details findet man in der Patran Online Dokumentation im „PCL and Customization“ Handbuch.

Am Beispiel des Darstellens des Materialnamens im Elementschwerpunkt selektierter Elemente soll die Funktionalität gezeigt werden. Im Folgenden wird auf Abschnitte des eigentlichen PCL Source Codes eingegangen.

`marker_color` legt die Farbe des Textobjektes fest
`seg_id` ist der Identifier des erzeugten Grafischen Segments

```
marker_color = 2  
gm_segment_create(seg_id)
```

Nachfolgende Auszüge werden für jedes Element innerhalb einer FOR-Schleife ausgeführt.

Der folgende Auszug ermittelt den Materialnamen, der als Text später den eigentlichen Inhalt unseres Textobjektes bildet.

```
/* Determines the current material name */  
db_get_material_name_from_id(mat_id, mat_name)
```

Hier bestimmen wir den jeweiligen Elementschwerpunkt, der als Position unseres Textobjektes dient.

```
/* Extracts element center locations */  
fem_geom_elem_location(melm_str, xyz_array, nextracted)
```

Umspeichern des 2dim Koordinatenarrayoutputs in einen Koordinatenvektorinput.

```
xyz(1) = xyz_array(1, 1)  
xyz(2) = xyz_array(1, 2)  
xyz(3) = xyz_array(1, 3)
```

Hier wird das jeweilige Textobjektes erzeugt.

```
gm_draw_text(seg_id, marker_color, xyz, mat_name)
```

Alle Grafischen Segmente werden mit Hilfe des nachfolgenden Befehls dargestellt.

```
gm_segment_flush()
```



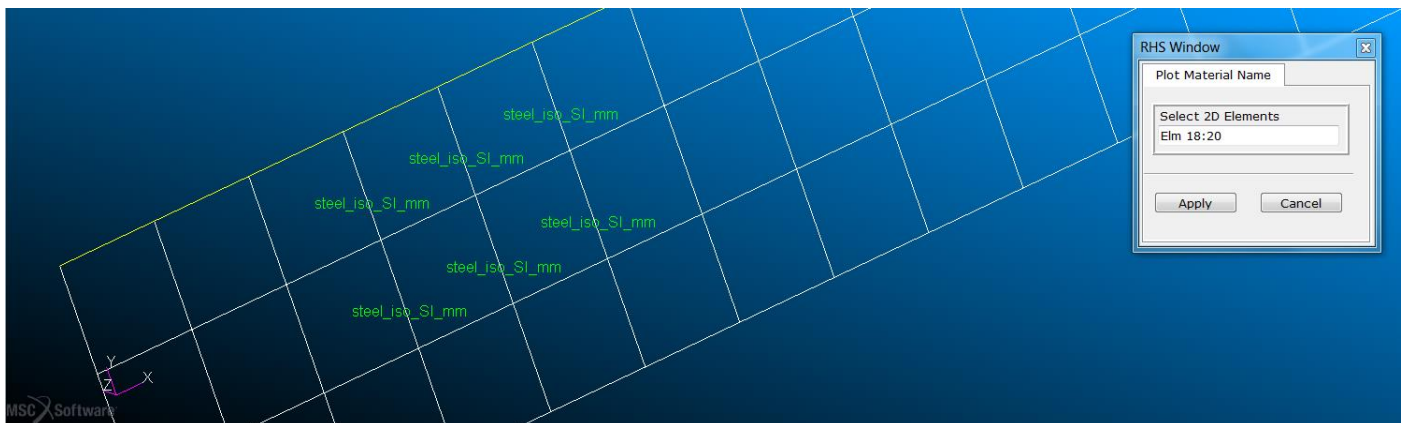
Patran

Grafische Basisobjekte in den Patran Viewport einfügen

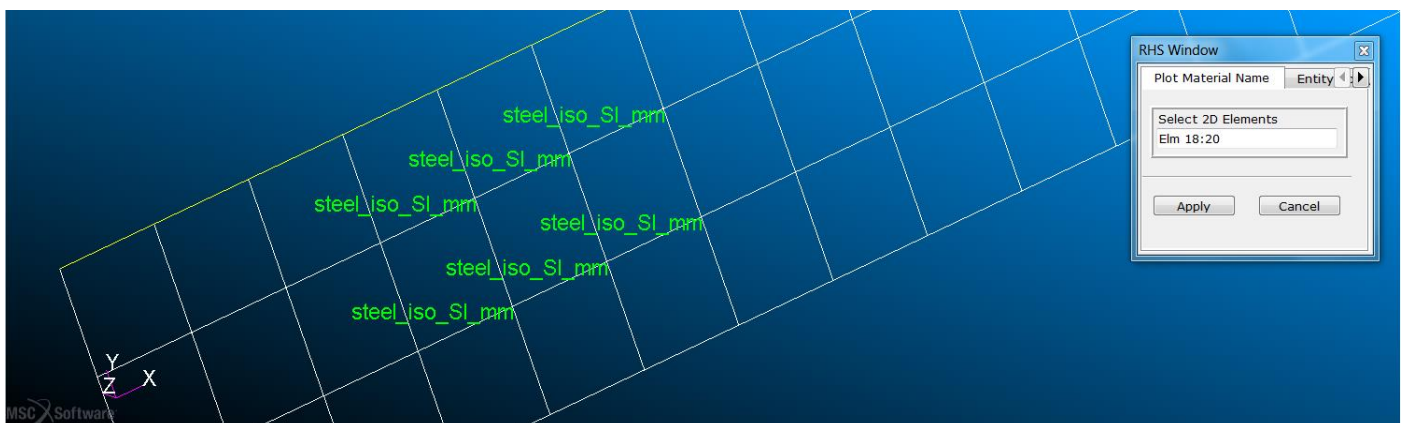
Beachtet werden sollte, daß man die ID der Grafiksegmente nicht ohne weiteres bestimmen kann. Daher hat sich folgender „Trick“ bewährt. Man erzeugt ein neues Grafisches Segment und löscht anschließend alle Grafischen Segmente mit kleineren IDs inklusive des neu erzeugten Grafischen Segments.

```
INTEGER seg_id, i
gm_segment_create(seg_id)
FOR (i = 1 TO seg_id)
    gm_segment_delete(i)
END FOR
```

PCL GUI und Patran Viewport:



Die Schriftgröße der Grafikschriften kann nachträglich über die Patran Standardfunktionalität im „Display“-Menü geändert werden. Hierzu wählt man einen der vorgegebenen Werte für „Label Font Size“ unter „Entity Color/Label/Render ...“.





Patran

Grafische Basisobjekte in den Patran Viewport einfügen

Der vollständige PCL Source Code wird nun aufgeführt.

```
#include "appforms.p"
#include "lpenums.p"
#include "lpkeywords.i"

#define CLASS_NAME          he_plot_mat_marker
#define CLASS_NAME_QUOTED  "he_plot_mat_marker"

CLASS CLASS_NAME

/* ABSTRACT: This program creates text segments at the center of elements
 * ===== showing the associated material name.
 *
 * KEYWORDS: None
 *
 * INPUT:  None
 *
 * OUTPUT: None
 *
 * RETURN VALUES: None
 *
 * SIDE EFFECTS:
 *
 * DESCRIPTION:
 * =====
 *
 * =====
 * REVISION HISTORY:
 *
 * REVISION DATE          AUTHOR          EMAIL
 * -----
 * 1          13-Jun-2017 Holger Eß / MSC Germany  holger.ess@mscsoftware.com
 *
 * REVISION number and reason
 * -----
 * 1  Creation.
 *
 */

          /* Variable initialization */

CLASSWISE WIDGET  main_form, sep, apply_button, cancel_button
CLASSWISE WIDGET  select_frame, select_dbox

CLASSWISE STRING  pick_list[VIRTUAL]
CLASSWISE STRING  message[512]

CLASSWISE REAL    form_height, form_width
```



Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```

/*-----*/

FUNCTION init()

/* ABSTRACT: This function initializes the main form.
*
* KEYWORDS: FORM, WIDGET, MAIN
*
* INPUT:  None
*
* OUTPUT: None
*
* RETURN VALUES: None
*
* SIDE EFFECTS: Hopefully none
*
* DESCRIPTION:
*
* =====
* REVISION HISTORY:
*
* REVISION DATE          AUTHOR                      EMAIL
* -----
* 1      13-Jun-2017 Holger Eß / MSC Germany  holger.ess@mscsoftware.com
*
* REVISION number and reason
* -----
* 1      Creation.
*
*/

REAL  y_loc, x_loc
REAL  sframe_height, sframe_width, sdbox_width, y_frame

form_height = FORM_HGT_FULLL
form_width  = FORM_WID_SML
sframe_height = SDBOX_HGT_LABOVE + SFRAME_T_MARGIN + SFRAME_B_MARGIN @
              + SFRAME_2EDGE
sframe_width  = SFRAME_WID_SINGLE
sdbox_width   = SDBOX_WID_SINGLE - SFRAME_2EDGE

x_loc = 0.5*(form_width - sframe_width)
y_loc = FORM_T_MARGIN + 1.0*INTER_WIDGET_SPACE

main_form=ui_form_create(                                     @
/* callback      */  "",                                     @
/* x             */  FORM_X_LOC_SML,                         @
/* y             */  FORM_Y_LOC,                             @
/* position      */  "UL",                                   @
/* width         */  form_width,                             @

```



Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```
/* height      */ form_height,      @
/* label       */ "Plot Material Name", @
/* unused      */ ""

/* This frame contains the select databox. */
select_frame = ui_selectframe_create( @
/* parent      */ main_form,        @
/* callback    */ "",               @
/* x           */ x_loc,            @
/* y           */ y_loc,            @
/* width       */ sframe_width,     @
/* height      */ sframe_height,    @
/* label       */ "",               @
/* recycle     */ TRUE)

y_frame = SFRAME_T_MARGIN

/* With help of that select databox the user can select the entities. */
select_dbox=ui_selectdatabox_create( @
/* parent      */ select_frame,     @
/* callback    */ "",               @
/* x           */ SFRAME_L_MARGIN,   @
/* y           */ y_frame,          @
/* label_length */ 0.0,              @
/* box_length  */ sdbox_width,      @
/* label       */ "Select 2D Elements", @
/* value       */ "",               @
/* label_above */ TRUE,             @
/* datatype    */ "ELEM2D",         @
/* prompt      */ "")

y_loc += sframe_height + 2*INTER_WIDGET_SPACE

sep = ui_separator_create( @
/* parent      */ main_form,        @
/* name        */ "",               @
/* x           */ 0,                @
/* y           */ y_loc,            @
/* width       */ form_width,       @
/* horizontal  */ TRUE)

y_loc += LINE_THICKNESS + 1.5 * INTER_WIDGET_SPACE

/* This button executes the request. */
apply_button = ui_button_create( @
/* parent      */ main_form,        @
/* callback    */ "apply_button_cb", @
/* x           */ BUTTON_HALF_X_LOC1, @
/* y           */ y_loc,            @
/* width       */ BUTTON_WID_HALF,   @
/* height      */ 0,                @
/* label       */ "Apply",           @
/* labelinside */ TRUE,              @
```



Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```

    /* highlight */ TRUE)

/* This button hides the main form. */
cancel_button = ui_button_create(
/* parent */ main_form,
/* callback */ "cancel_cb",
/* x */ BUTTON_HALF_X_LOC2,
/* y */ y_loc,
/* width */ BUTTON_WID_HALF,
/* height */ 0,
/* label */ "Cancel",
/* labelinside */ TRUE,
/* highlight */ FALSE)

y_loc += BUTTON_DEFAULT_HGT + 1.0 * INTER_WIDGET_SPACE

form_height = y_loc + FORM_B_MARGIN
ui_wid_set(main_form,"HEIGHT", form_height)

END FUNCTION /* init */

/*-----*/

FUNCTION display()

/* ABSTRACT: This function controls the display of the GUI.
* =====
*
* KEYWORDS: DISPLAY, FORM
*
* INPUT: None
*
* OUTPUT: None
*
* RETURN VALUES: 0 finished successful
*
* SIDE EFFECTS:
*
* DESCRIPTION:
* =====
*
* =====
* REVISION HISTORY:
*
* REVISION DATE AUTHOR EMAIL
* -----
* 1 13-Jun-2017 Holger Eß / MSC Germany holger.ess@mscsoftware.com
*
* REVISION number and reason
* -----

```



Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```

* 1 Creation.
*
*/

/* Check to see if Main Form has been initialized */
IF (!ui_form_exists(CLASS_NAME_QUOTED)) THEN
    ui_exec_function(CLASS_NAME_QUOTED, "init")
ENDIF

ui_form_display(CLASS_NAME_QUOTED)

ui_wid_set(select_dbox, "VALUE", "")
ui_set_focus(select_dbox)
ui_wid_refresh()

RETURN(0)

END FUNCTION /* display */

/*-----*/

FUNCTION apply_button_cb()

/* ABSTRACT: This callback is associated to the button apply_button.
* =====
*
* KEYWORDS: CALLBACK, WIDGET, BUTTON
*
* INPUT:      None
*
* OUTPUT:     None
*
* RETURN VALUES: 0      finished successful
*
* SIDE EFFECTS:
*
*
* DESCRIPTION:
* =====
*
* =====
* REVISION HISTORY:
*
* REVISION DATE          AUTHOR                      EMAIL
* -----
* 1      13-Jun-2017 Holger Eß / MSC Germany  holger.ess@mscsoftware.com
*
* REVISION number and reason
* -----
* 1 Creation.
*
*/

```




Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```
INTEGER    status
INTEGER    nelem_ids, elem_ids(VIRTUAL), region_ids(VIRTUAL), mat_id, seg_id
INTEGER    i, i_01, i_02, i_03, i_04, i_05, i_06, i_07, i_09, i_10
INTEGER    marker_color, nextracted

REAL       xyz_array(VIRTUAL), xyz(3)

STRING     mat_name[32], s_01[32], melm_str[VIRTUAL]

/* Sets the autofitting, autocentering and autoextending to FALSE */
auto_fitview(FALSE)
auto_center(FALSE)
auto_extend(FALSE)

ui_wid_get_vstring(select_dbox, "VALUE", pick_list)

/* In case of an empty pick list a warning message appears. */
IF (pick_list == " ") THEN
    message = "No 2D Elements selected!"
    write_line(message)

    RETURN(-1)
ENDIF

marker_color = 2
gm_segment_create(seg_id)

/* Counts the number of elements of the select string */
nelem_ids = fem_u_count_id_list(LP_SUBLIST_ELEMENT, @
                                pick_list,          @
                                TRUE,                @
                                status)

SYS_ALLOCATE_ARRAY(elem_ids, 1, nelem_ids)
SYS_ALLOCATE_ARRAY(region_ids, 1, nelem_ids)

/* Gets the element IDs of the select string */
fem_u_get_id_list(LP_SUBLIST_ELEMENT, @
                 pick_list,          @
                 nelem_ids,         @
                 TRUE,               @
                 elem_ids)

SYS_ALLOCATE_STRING(melm_str, 7000)

/* Determines the Property Region IDs of the selected elements */
db_get_region_for_elements(nelem_ids, elem_ids, region_ids)

FOR (i = 1 TO nelem_ids) LOOP_ELEMS
    IF (region_ids(i) != 0) THEN
/* Determines the Property Region Definition of the current element */
        status = db_get_region_definition(region_ids(i), @
```




Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```

                                s_01,          @
                                i_01,          @
                                i_02,          @
                                i_03,          @
                                i_04,          @
                                i_05,          @
                                i_06,          @
                                i_07,          @
                                mat_id,       @
                                i_09,          @
                                i_10)

IF (status != 0) THEN
    msg_to_form(status, 4, appcode(status), 1, 1.0, "")
    RETURN(status)
ENDIF

IF (mat_id != 0) THEN
/* Determines the current material name */
    db_get_material_name_from_id(mat_id, mat_name)

    melm_str = "Elem"//" " //str_from_integer(elem_ids(i))

/* Extracts element center locations */
    nextracted = 0
    fem_geom_elem_location(melm_str, xyz_array, nextracted)
    IF (nextracted == 0) THEN
        RETURN(-2)
    ENDIF

    xyz(1) = xyz_array(1, 1)
    xyz(2) = xyz_array(1, 2)
    xyz(3) = xyz_array(1, 3)

    SYS_FREE_ARRAY(xyz_array)

    gm_draw_text(seg_id, marker_color, xyz, mat_name)
ENDIF
END FOR

gm_segment_flush()

RETURN(0)

END FUNCTION /* apply_button_cb */

/*-----*/

FUNCTION cancel_cb()

/* ABSTRACT: This callback is associated to the button cancel_button.
 * =====
 *

```



Patran

Grafische Basisobjekte in den Patran Viewport einfügen

```

* KEYWORDS: CALLBACK, WIDGET, BUTTON
*
* INPUT:      None
*
* OUTPUT:     None
*
* RETURN VALUES: 0      finished successful
*
* SIDE EFFECTS:
*
*
* DESCRIPTION:
* =====
*
* =====
* REVISION HISTORY:
*
* REVISION DATE          AUTHOR                      EMAIL
* -----
* 1      13-Jun-2017 Holger Eß / MSC Germany  holger.ess@mscsoftware.com
*
* REVISION number and reason
* -----
* 1      Creation.
*
*/

INTEGER seg_id, i

  gm_segment_create(seg_id)
  FOR (i = 1 TO seg_id)
    gm_segment_delete(i)
  END FOR

  display_cleanup( )

  select_focus.exit()
  ui_form_hide(CLASS_NAME_QUOTED)

  RETURN(0)

END FUNCTION /* cancel_cb */

/*-----*/

END CLASS /* he_plot_mat_marker */

```