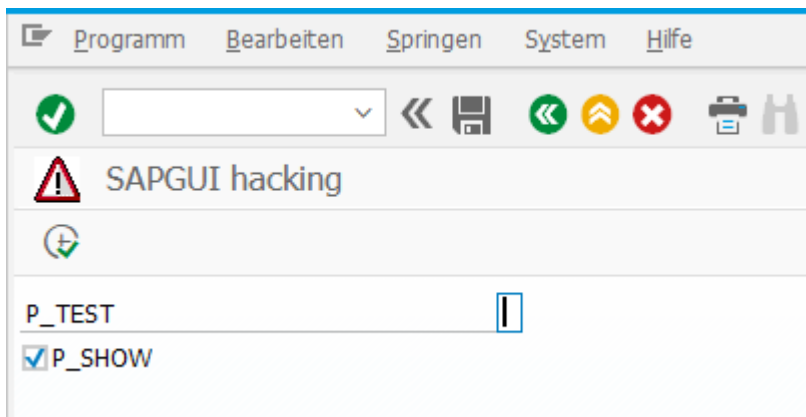


Hacking SAPGUI

Heute bin ich zufällig auf etwas gestoßen, dass ich erst nicht glauben konnte. Aber eigentlich hätte es klar sein müssen. Ich zeige dir erst einmal ein Bild:



Icon an ungewöhnlicher Stelle...

Vielleicht ist deine erste Reaktion genauso wie meine:

[via GIPHY](#)

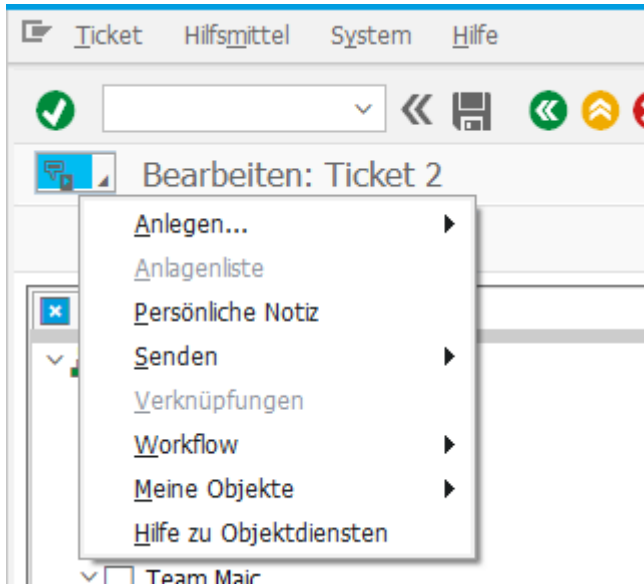
Folgende Controls verwende ich für diesen kleinen Hack:

- CL_GUI_PICTURE
- CL_GUI_GOS_CONTAINER

Normalerweise wird der GOS-Container nicht explizit aufgerufen, sondern nur implizit von der Klasse CL_GOS_MANAGER verwendet.

Container ist Container

Im Generic Object Services Menü wird normalerweise ein Pull-Down-Menü angezeigt:



Das GOS-Menü in Aktion

In einem Menü können jedoch alle Controls angezeigt werden. Sinnvoll sind hier nur wenige, denn der GOS-Container ist ziemlich klein. Ein Menü passt dort gut hinein. Aber auch ein Bild oder Icon.

Anzeige Icon in Container

Die Anzeige eines Bildes oder Icons ist ziemlich simpel und schnell erledigt:

REPORT.

PARAMETERS p_test.

INITIALIZATION.

```

DATA(picture) = NEW cl_gui_picture( parent = NEW cl_gui_gos_container(
width = 38 ) ).
DATA url TYPE cndp_url.
CALL FUNCTION 'DP_PUBLISH_WWW_URL'
  EXPORTING
    objid          = 'ACHTUNG'
    lifetime       = cndp_lifetime_transaction
  IMPORTING
    url            = url
  EXCEPTIONS
    dp_invalid_parameters = 1
    no_object           = 2
    dp_error_publish    = 3.
IF sy-subrc = 0.

```

```
picture->load_picture_from_url_async( url = url ).
picture->set_display_mode( picture->display_mode_fit ).
ENDIF.
```

Da geht noch mehr...

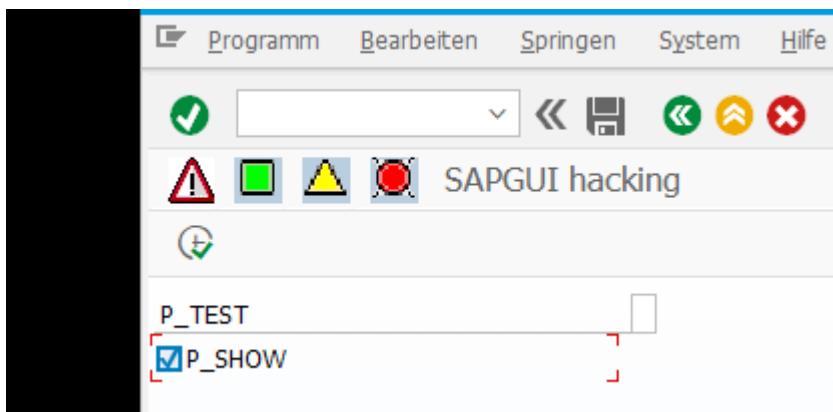
Nachdem ich ein bisschen herumgespielt habe, ist mir eine Eigenschaft aufgefallen, die nicht ganz offensichtlich ist und die ich auch so nicht erwartet hätte. So, wie man mehrere Docking-Container an ein Dynpro andocken kann, so kann man auch den CL_GUI_GOS_CONTAINER mehrfach erzeugen.

Zusätzlich können in einem Picture-Control auch die Klick- und Doppelklick-Ereignisse aktiviert und verwendet werden.

Auch das Ein- und Ausblenden des Controls ist möglich (Methode SET_VISIBLE).

Code

Folgendes kleine Programm zeigt die Möglichkeiten, die sich ergeben:



Mehrere GOS-Container

Durch Klicken des Parameters P_SHOW wird das Bild „ACHTUNG“ ein- und ausgeblendet. Ein Klick auf eines der Icons zeigt eine Info-Meldung. Ich nutze solch kleine Spielereien immer gerne, um mich an die neue Syntax zu gewöhnen und auszutesten, was möglich und sinnvoll ist. Hier habe ich die Gelegenheit am Schopfe gepackt und auch funktional programmiert, so dass Method-Chaining über mehrere Methoden hinweg auf ein und das selbe Objekt möglich ist.

REPORT.

PARAMETERS p_test.

PARAMETERS p_show AS CHECKBOX DEFAULT 'X' USER-COMMAND dummy.

CLASS info DEFINITION.

PUBLIC SECTION.

METHODS icon IMPORTING name TYPE clike RETURNING VALUE(info) TYPE REF TO info.

METHODS pic IMPORTING name TYPE clike RETURNING VALUE(info) TYPE REF TO info.

METHODS constructor IMPORTING width TYPE i.

METHODS with_text IMPORTING text TYPE clike RETURNING VALUE(info) TYPE REF TO info..

METHODS hide.

METHODS show.

PROTECTED SECTION.

DATA picture TYPE REF TO cl_gui_picture.

METHODS handle_click FOR EVENT picture_click OF cl_gui_picture.

DATA text TYPE string.

ENDCLASS.

CLASS info IMPLEMENTATION.

METHOD constructor.

picture = NEW cl_gui_picture(parent = NEW cl_gui_gos_container(width = width)).

picture->set_registered_events(VALUE #((eventid = cl_gui_picture=>eventid_picture_click))).

SET HANDLER handle_click FOR picture.

ENDMETHOD.

METHOD icon.

picture->load_picture_from_sap_icons(name).

picture->set_display_mode(picture->display_mode_fit).

info = me.

ENDMETHOD.

METHOD pic.

DATA url TYPE cndp_url.

CALL FUNCTION 'DP_PUBLISH_WWW_URL'

EXPORTING

objid = CONV w3objid(name)

lifetime = cndp_lifetime_transaction

IMPORTING

url = url

EXCEPTIONS

dp_invalid_parameters = 1

no_object = 2

dp_error_publish = 3.

IF sy-subrc = 0.

```

        picture->load_picture_from_url_async( url = url ).
        picture->set_display_mode( picture->display_mode_fit ).
    ENDIF.

    info = me.

ENDMETHOD.

METHOD with_text.
    me->text = text.
    info = me.
ENDMETHOD.

METHOD handle_click.
    CHECK text IS NOT INITIAL.
    MESSAGE text TYPE 'I'.
ENDMETHOD.

METHOD hide.
    picture->set_visible( space ).
ENDMETHOD.
METHOD show.
    picture->set_visible( 'X' ).
ENDMETHOD.

ENDCLASS.

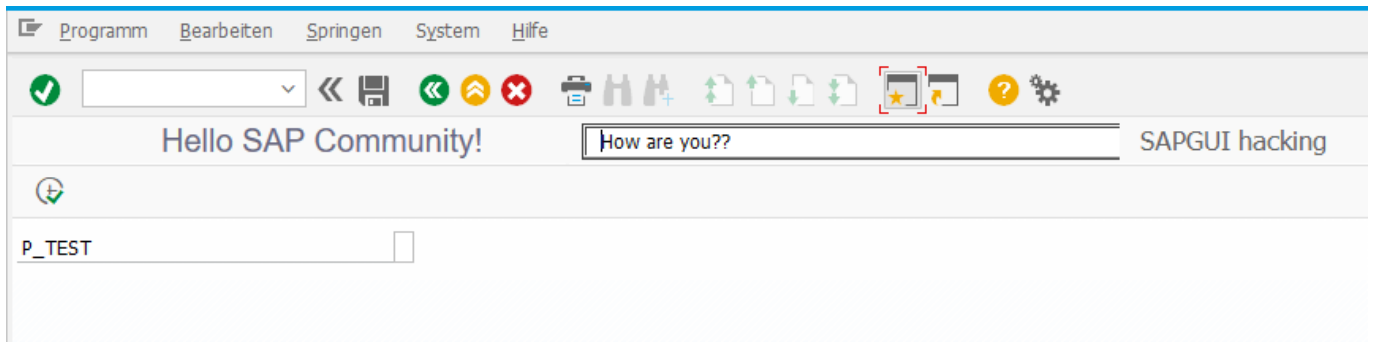
INITIALIZATION.
    DATA(info)    = NEW info( 38 )->pic( 'ACHTUNG' ).
    DATA(green)  = NEW info( 39 )->icon( icon_led_green )->with_text( 'Alles
ok' ).
    DATA(yellow) = NEW info( 39 )->icon( icon_led_yellow )->with_text(
'hmpffffff' ).
    DATA(red)    = NEW info( 39 )->icon( icon_led_red )->with_text( 'error.
error. error.' ).

AT SELECTION-SCREEN OUTPUT.
CASE p_show.
    WHEN abap_true.
        info->show( ).
    WHEN abap_false.
        info->hide( ).
ENDCASE.

```

WTF?!

Man kann übrigens jedes Control in den Container packen! Auch Text-Edit-Controls und HTML-Viewer... ☐



REPORT.

PARAMETERS p_test.

INITIALIZATION.

```
DATA(html) = NEW cl_gui_html_viewer( parent = NEW cl_gui_gos_container(
width = 400 ) ).
```

```
DATA url TYPE c LENGTH 100.
```

```
DATA data TYPE STANDARD TABLE OF w3_html.
```

```
data = VALUE #( ( '<html><head><style>body { margin: 0; background-color:
#f9f9f9; color: #666680; font: 20px "Arial" }</style>'
&& '<title>Hacking SAPGUI</title></head><body
background=#aabbcc><marquee
behavior=alternate>tricktresor.de</marquee></body></html>' ) ).
```

```
html->load_data( IMPORTING assigned_url = url
CHANGING data_table = data ).
```

```
html->show_url( url ).
```

```
DATA(text) = NEW cl_gui_textedit( parent = NEW cl_gui_gos_container( width
= 400 ) ).
```

```
text->set_statusbar_mode( 0 ).
```

```
text->set_toolbar_mode( 0 ).
```

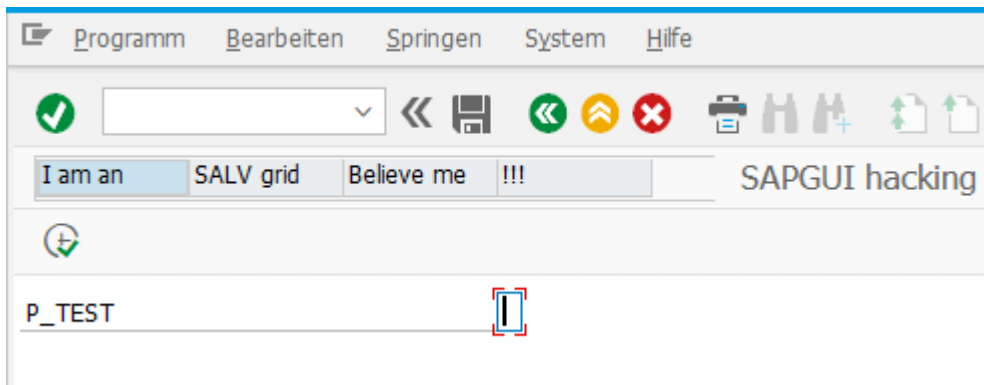
```
text->set_textstream( `Enter your name` ).
```

```
text->select_lines( 1 ).
```

```
cl_gui_control=>set_focus( text ).
```

SALV-Grid

Für ein ALV ist sicherlich nicht genug Platz...? Denkste:



CL_SALV_TABLE in CL_GUI_GOS_CONTAINER