Hierarchiedarstellung von Controls

Hierarchien sind was Tolles! Controls sind auch toll! Wir sollten sie nutzen, so lange es sie noch gibt (Stichwort Fiori, ABAP in the Cloud etc.). Deswegen ein schönes Demoprogramm zur Analyse von GUI-Controls zur Laufzeit und Darstellung als Baumstruktur mit Hilfe der Klasse CL_COLUMN_TREE_MODEL.

GUI-Elemente

GUI-Controls sind ActiveX-Steuerlemente, die mit dem Stichwort EnjoySAP zu SAP Release 4.6C eingeführt wurden. Sie werden auch OCX-Controls genannt. Hier gibt es im Grunde zwei unterschiedliche Arten:

- Container
- Controls

Sie alle haben eine Gemeinsamkeit, sie erben nämlich alle von der Hauptklasse CL_GUI_CONTROL. Verwirrender Weise erbt CL_GUI_CONTAINER von CL_GUI_CONTROL.

Container

Typische Container sind:

- CL_GUI_CUSTOM_CONTAINER Container for customer controls in the dynpro area
- CL_GUI_DIALOGBOX_CONTAINER Container for customer controls in the dynpro area
- CL_GUI_DOCKING_CONTAINER Docking Control Container
- CL_GUI_EASY_SPLITTER_CONTAINER Reduced Version of Splitter Container Control
- CL_GUI_SIMPLE_CONTAINER Anonymous Container

- CL_GUI_SPLITTER_CONTAINER Splitter Control

Etwas ungewöhnlicher ist der CL_GUI_GOS_CONTAINER für Generic Objects. Wer sich ansehen möchte, was für Schweinereien man mit diesem machen kann, sollte sich Hacking SAPGUI ansehen.

Container haben die Besonderheit, dass sie weitere Container oder Controls enthalten können.

Controls

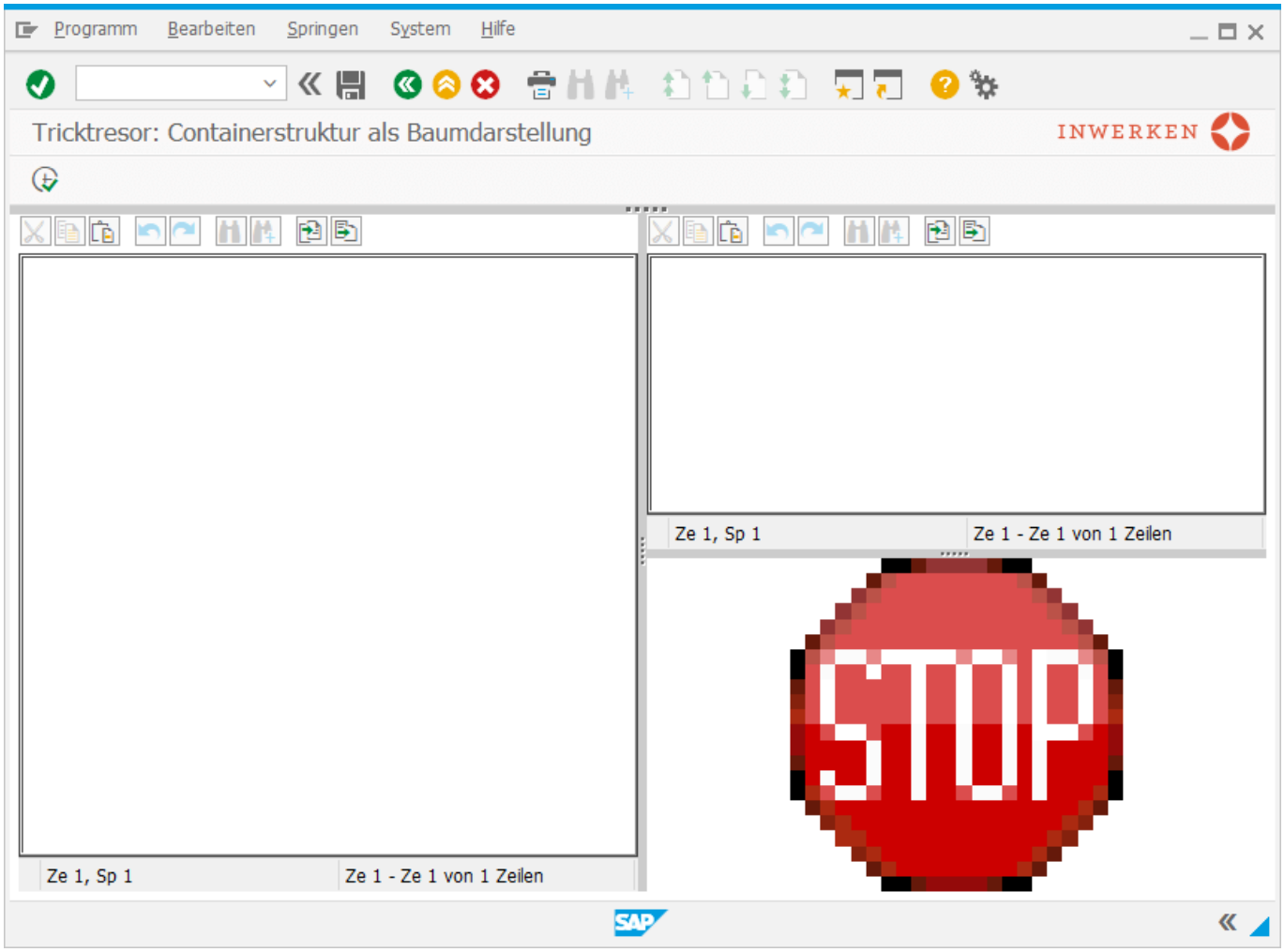
Die meistgebrauchten Controls sind sicherlich:

- CL_GUI_ALV_GRID
- CL_GUI_TEXTEDIT
- CL_GUI_PICTURE
- CL_GUI_HTML_VIEWER
- CL_GUI_TOOLBAR

Sie alle benötigen einen CL_GUI_CONTAINER, in dem sie dargestellt werden.

Control-Hierarchie

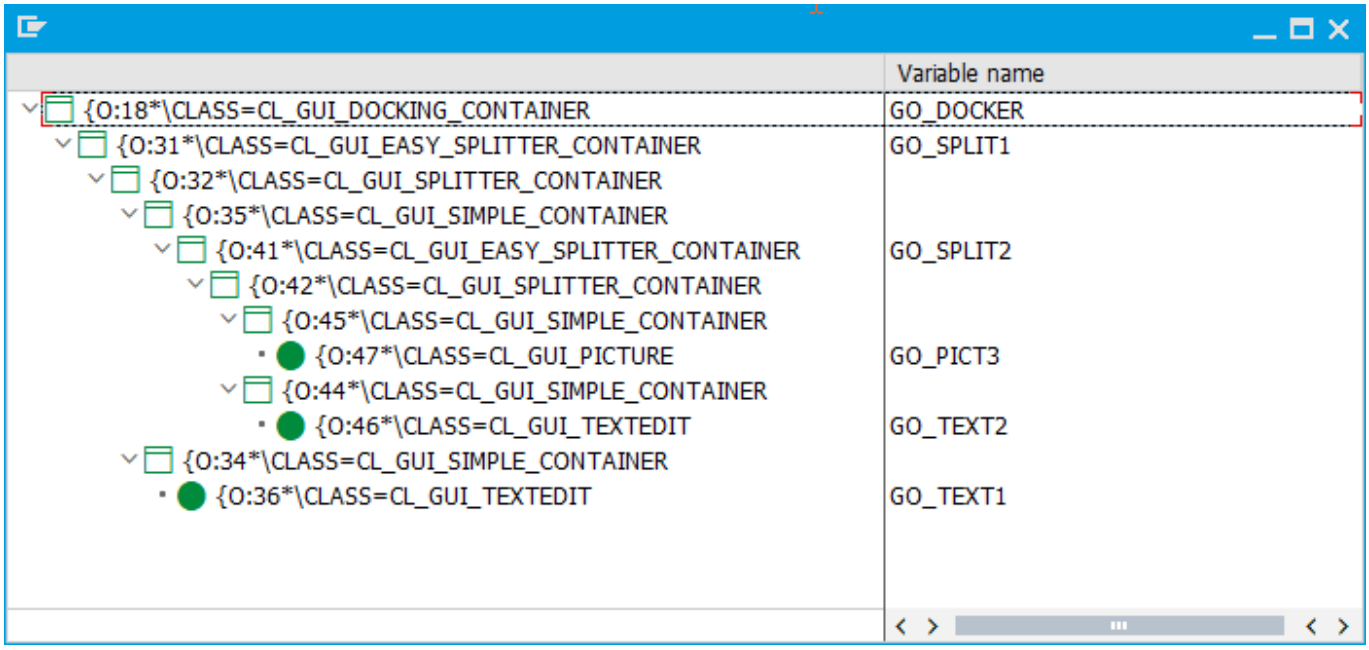
Ich habe ein kleines Programm geschrieben, das eine einfache Anwendung aus zwei Splittern, zwei Textedit-Controls und einem Picture-Control erstellt.



Container und Controls

Hierarchieanalyse

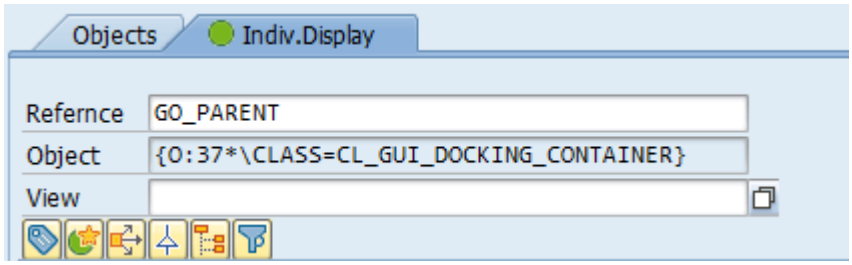
Die Klasse ZCL_TRCKTRSR_CONTAINER_TREE analysiert die Objektstruktur und stellt diese hierarchisch dar.



Hierarchiedarstellung der verwendeten GUI-Controls

Objekt-ID

Im Debugger sieht man die Objekt-ID der erzeugten Objekte:



Objekt-ID von GO_PARENT

Die Objekte werden bei Programmausführung durchnummeriert und erhalten so eine eindeutige Objekt-ID. Der folgende Code ermittelt die Objekt-ID zu einem Objekt:

```
DATA lo_obj TYPE REF TO cl_abap_objectdescr.

lo_obj ?= cl_abap_typedescr=>describe_by_object_ref( io_object ).

DATA(lv_relname) = lo_obj->get_relative_name( ).
DATA lv_object_id TYPE i.
CALL 'OBJMGR_GET_INFO' ID 'OPNAME' FIELD 'GET_OBJID'
                       ID 'OBJID'  FIELD lv_object_id
                       ID 'OBJ'    FIELD io_object.
```

```
rv_object_name = |\{O:{ lv_object_id }*{
                  lo_obj->absolute_name }|.
```

github

Das Demoprogramm liegt bei github und kann mit Hilfe von abapGit heruntergeladen werden:

https://github.com/tricktresor/container_hierarchy

guidrasil

Wenn du Lust bekommen hast, dich noch weiter mit Controls zu beschäftigen, dann schaue dir meinen Control-Designer guidrasil an.