

SAP Container



Release 4.6C



Copyright

© Copyright 2001 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Software-Produkte können Software-Komponenten auch anderer Software-Hersteller enthalten.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] und SQL Server[®] sind eingetragene Marken der Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®] und OS/400[®] sind eingetragene Marken der IBM Corporation.

ORACLE[®] ist eine eingetragene Marke der ORACLE Corporation.

INFORMIX[®]-OnLine for SAP und Informix[®] Dynamic Server[™] sind eingetragene Marken der Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®] und Motif[®] sind eingetragene Marken der Open Group.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT[®] ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo und mySAP.com sind Marken oder eingetragene Marken der SAP AG in Deutschland und vielen anderen Ländern weltweit. Alle anderen Produkte sind Marken oder eingetragene Marken der jeweiligen Firmen.

Symbole

| Symbol | Bedeutung |
|---|------------|
|  | Achtung |
|  | Beispiel |
|  | Empfehlung |
|  | Hinweis |
|  | Syntax |
|  | Tip |

Inhalt

| | |
|---|-----------|
| SAP Container | 6 |
| Visualisierung und Re-link von Containern | 7 |
| Methoden | 8 |
| CLASS_CONSTRUCTOR | 9 |
| CONSTRUCTOR..... | 10 |
| LINK..... | 12 |
| Instanz-Attribute | 13 |
| Statische Attribute..... | 14 |
| SAP Custom Container | 15 |
| CONSTRUCTOR..... | 16 |
| SAP Dialogbox Container | 18 |
| Methoden..... | 19 |
| CONSTRUCTOR..... | 20 |
| SET_CAPTION..... | 22 |
| Ereignisse..... | 23 |
| CLOSE..... | 24 |
| SAP Docking Container | 26 |
| Methoden..... | 27 |
| CONSTRUCTOR..... | 28 |
| DOCK_AT..... | 31 |
| FLOAT | 32 |
| GET_EXTENSION | 33 |
| SET_CAPTION..... | 34 |
| SET_EXTENSION..... | 35 |
| Statische Konstanten | 36 |
| SAP Splitter Container | 37 |
| Methoden..... | 38 |
| ADD_CONTROL | 39 |
| ADD_CONTROL_HANDLE..... | 40 |
| CONSTRUCTOR..... | 41 |
| GET_CONTAINER | 43 |
| GET_COLUMNS | 44 |
| GET_COLUMN_MODE..... | 45 |
| GET_COLUMN_WIDTH..... | 46 |
| GET_COLUMN_SASH..... | 47 |
| GET_ROWS | 48 |
| GET_ROW_HEIGHT | 49 |
| GET_ROW_MODE..... | 50 |
| GET_ROW_SASH..... | 51 |
| REMOVE_CONTROL | 52 |
| SET_BORDER | 53 |
| SET_COLUMN_MODE | 54 |
| SET_COLUMN_WIDTH..... | 55 |
| SET_COLUMN_SASH | 56 |
| SET_GRID..... | 57 |
| SET_ROW_HEIGHT | 58 |

| | |
|---|-----------|
| SET_ROW_MODE | 59 |
| SET_ROW_SASH | 60 |
| Statische Konstanten | 61 |
| SAP Easy Splitter Container | 62 |
| Methoden | 63 |
| CONSTRUCTOR | 64 |
| SET_SASH_POSITION | 66 |
| Statische Attribute | 67 |
| Instanz-Attribute | 68 |
| Methoden der Klasse CL_GUI_CFW | 69 |
| dispatch | 70 |
| flush | 71 |
| get_living_dynpro_controls | 72 |
| set_new_ok_code | 73 |
| update_view | 74 |
| Methoden der Klasse CL_GUI_CONTROL | 75 |
| constructor | 76 |
| finalize | 78 |
| get_focus | 79 |
| get_height | 80 |
| get_registered_events | 81 |
| get_width | 82 |
| is_alive | 83 |
| set_alignment | 84 |
| set_focus | 85 |
| set_position | 86 |
| set_registered_events | 87 |
| set_visible | 88 |
| Methoden der Klasse CL_GUI_OBJECT | 89 |
| free | 90 |
| is_valid | 91 |

SAP Container

Verwendung

Ein SAP Container ist in der Lage, andere Controls (z.B. SAP Tree Control, SAP Picture Control, SAP TextEdit Control, SAP Splitter Control, usw.) zu beherbergen. Es verwaltet diese Controls logisch in einer Collection und sorgt für einen physischen Bereich zur Visualisierung.

Jedes Control lebt in einem Container. Da Container selbst Controls sind, können Container ineinander geschachtelt werden. Der Container wird zum Parent seines Controls.

Die SAP Container sind in fünf Gruppen unterteilt:

- [SAP Custom Container \[Seite 15\]](#)
Darstellung von Controls innerhalb eines im Screen Painter definierten Bereichs auf Dynpros oder Subscreens
Klasse: CL_GUI_CUSTOM_CONTAINER
- [SAP Dialogbox Container \[Seite 18\]](#)
Darstellung von Controls in einem amodalen Dialogfenster oder als Fullscreen
Klasse: CL_GUI_DIALOGBOX_CONTAINER
- [SAP Docking Container \[Seite 26\]](#)
Darstellung von Controls als gedocktes, resizebares Teilfenster mit der Option, dieses als eigenständiges amodales Dialogfenster darzustellen. Der Docking Container kann somit an die vier Ränder eines Dynpros gebunden und wieder getrennt werden
Klasse: CL_GUI_DOCKING_CONTAINER
- [SAP Splitter Container \[Seite 37\]](#)
Darstellung und Gruppierung von mehreren Controls in einen Bereich, d.h. Unterteilung des Bereichs in Zellen
Klasse: CL_GUI_SPLITTER_CONTAINER
- [SAP Easy Splitter Container \[Seite 62\]](#)
Darstellung von Controls in zwei Zellen mit einem verschiebbaren Trennbalken (Split Bar)
Klasse: CL_GUI_EASY_SPLITTER_CONTAINER

Alle SAP Container haben einen gemeinsamen Parent vom Typ CL_GUI_CONTROL. Sie leiten sich von diesem Basiscontainer ab und besitzen somit eine einheitliche objektorientierte Schnittstelle.

Visualisierung und Re-link von Containern

Visualisierung

Bei der Visualisierung von SAP Containern sind folgende Punkte zu beachten:

- Bei der Visualisierung ist die Defaultgröße der Controls gleich der Containergröße. Abweichungen hiervon können Sie beim Erzeugen der Instanz definieren.
- Jeder Container hat einen Top-Level Container (Screen0 bis 9) als Parent. Dieser Top-Level Container wird automatisch beim Aufruf der Klasse erzeugt und ist gleichbedeutend mit den Dynpros der Pop-up Levels 0 bis 9. Als Default Screen wird beim Erzeugen das gerade existierende Pop-up Level (Stack Level) benutzt. Falls Sie eine explizite Definition des Screens wünschen, übergeben Sie beim Erzeugen des Containers einen entsprechenden Screen als Parent (siehe entsprechende Konstruktoren).
- Die Visualisierung eines Containers ist auf das assoziierte Pop-up Level beschränkt. Beim Verlassen eines Levels werden alle Controls vom Lifetime Management abgebaut (Weitere Informationen finden Sie unter [Lifetime Management \[Extern\]](#)).
- Falls Sie Controls dynamisch anzeigen möchten, benutzen Sie die Methode [SET_VISIBLE \[Seite 88\]](#).
- Falls der Parent nicht sichtbar ist, sind ebenfalls seine Children nicht sichtbar.

Re-link

Das Re-Linken von Containern ist nur auf dem gleichen Pop-up Level und auf dem ersten (obersten) Parent Level (z.B. Custom Container, Docking Container) möglich.

Hierzu benutzen Sie die Methode [LINK \[Seite 12\]](#).

Methoden

Methoden

[CLASS_CONSTRUCTOR \[Seite 9\]](#)

[CONSTRUCTOR \[Seite 10\]](#)

[LINK \[Seite 12\]](#)

CLASS_CONSTRUCTOR

Verwendung

Diese Methode erzeugt benötigte Pop-up Levels für die SAP Container.



Da der Klassenkonstruktor beim ersten Zugriff auf ein Element der Klasse aufgerufen wird, rufen Sie diese Methode niemals explizit auf.

Weitere Informationen finden Sie unter [Visualisierung und Re-link von Containern \[Seite 7\]](#).

CONSTRUCTOR

CONSTRUCTOR

Verwendung

Der Konstruktor kreiert, initialisiert und bindet das Control.



Der Konstruktor wird automatisch bei der Erzeugung einer Instanz (**create object**) aufgerufen. Die Parameter übergeben Sie normalerweise bei diesem Aufruf. Da die SAP Container Klasse eine Superklasse ist, erfolgt der Aufruf indirekt über den Aufruf des Konstruktors in der jeweiligen Subklasse.

Funktionsumfang

create object container

```
exporting
  clsid = clsid
  parent = parent
  style = style
  dynnr = dynnr
  repid = repid
  container_name = container_name
  lifetime = lifetime
  autoalign = autoalign
  no_autodef_progid_dynnr = no_autodef_progid_dynnr
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2
  create_error = 3
  lifetime_error = 4
  lifetime_dynpro_dynpro_link = 5
  lifetime_dynpro_illegal_parent = 6.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|---|
| clsid | Klassen-ID | |
| parent | Parent der Instanz; d.h., Container in dem das Control angezeigt wird | |
| style | Steuerung des Erscheinungsbild und des Verhaltens des Controls | Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit WS_*) |
| dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| repid | Report-ID: Report, an den der Container gebunden werden soll | |

CONSTRUCTOR

| | | |
|-------------------------|--|---|
| container_name | Name des im Screen Painter definierten Custom Containers | |
| lifetime | Parameter des Lifetime Managements (Lebensdauer des Controls) | <p>cntl_lifetime_imode: Das Control lebt solange der interne Modus nicht abgebaut ist (z.B. <code>leave program</code>, <code>leave transaction</code>)</p> <p>cntl_lifetime_dynpro: Das Control lebt solange die Instanz des Dynpros existiert, d.h. noch im Dynprostapel vorhanden ist (z.B. <code>call screen 100</code>, <code>call transaction</code>)</p> |
| autoalign | automatisches Ausrichten der Container | Default: kein automatisches Ausrichten |
| no_autodef_progid_dynnr | automatische Definition der Programm-ID und der Dynpronummer ausschalten | <p>space Programm-ID und Dynpronummer werden automatisch gezogen (Defaultwert)</p> <p>'x' automatische Definition der Programm-ID und der Dynpronummer ausgeschaltet</p> |

LINK

LINK

Verwendung

Mit dieser Methode verbinden Sie Controls mit einem neuen Custom Container. Die Referenzvariable verweist jetzt auf den neuen Custom Container, d.h. das Control wird im vorherigen Custom Container nicht mehr angezeigt.



Das Re-Linken von Containern ist nur auf dem gleichen Pop-up Level und auf dem ersten (obersten) Parent Level (z.B. Custom Container, Docking Container) möglich.

Funktionsumfang

call method container->link

exporting

repid = repid

dynnr = dynnr

container = container

exceptions

cntl_error = 1

cntl_system_error = 2

lifetime_dynpro_dynpro_link = 3.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|----------------|
| dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| repid | Report-ID: Report, an dem das Control gebunden werden soll | |
| container | Name des im Screen Painter definierten Custom Containers | |

Instanz-Attribute

| Instanz-Attribut | Type / Value | Beschreibung / Verwendung |
|------------------|-------------------|---------------------------|
| children | CNTO_CONTROL_LIST | Liste der Children |

Statische Attribute

Statische Attribute

| Statisches Attribut | Type / Value | Beschreibung / Verwendung |
|---------------------|------------------|---|
| default_screen | CL_GUI_CONTAINER | Dummy für Default Top-Level Container (Screen) |
| screen0 bis screen9 | CL_GUI_CONTAINER | Dummy für Default Top-Level0 bis Top-Level9 Container (Screen0 bis Screen9) |

SAP Custom Container

Verwendung

Mit Hilfe des SAP Custom Containers binden Sie in einen Bereich Ihres Dynpros oder Subscreens Controls ein. Die Definition des Bereiches auf Ihrem Dynpro erfolgt im Screen Painter.

Die eingebundenen Controls werden als Default auf die Größe des Containers gesetzt.

Voraussetzungen


Vor dem Einbinden eines Controls in den Custom Container müssen Sie im Screen Painter einen Bereich auf Ihrem Dynpro definiert haben. Weitere Informationen finden Sie unter [Custom Control anlegen \[Extern\]](#).

CONSTRUCTOR

CONSTRUCTOR

Verwendung

Der Konstruktor kreiert, initialisiert und bindet das Control.

 Der Konstruktor wird automatisch bei der Erzeugung einer Instanz (`create object`) aufgerufen. Die Parameter übergeben Sie normalerweise bei diesem Aufruf.

Funktionsumfang

create object custom_container

```
exporting
  parent = parent
  container_name = container_name
  style = style
  dynnr = dynnr
  repid = repid
  lifetime = lifetime
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2
  create_error = 3
  lifetime_error = 4
  lifetime_dynpro_dynpro_link = 5.
```

| Parameter | Bedeutung | Mögliche Werte |
|----------------|---|---|
| parent | Parent der Instanz; d.h., Container in dem das Control angezeigert wird | |
| container_name | Name des im Screen Painter definierten Custom Containers | |
| style | Steuerung des Erscheinungsbild und des Verhaltens des Controls | Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit WS_*) |
| dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| repid | Report-ID: Report, an den der Container gebunden werden soll | |

CONSTRUCTOR

| | | |
|-------------------------|--|---|
| lifetime | Parameter des Lifetime Managements (Lebensdauer des Controls) | <p>cntl_lifetime_imode: Das Control lebt solange der interne Modus nicht abgebaut ist (z.B. leave program, leave transaction)</p> <p>cntl_lifetime_dynpro: Das Control lebt solange die Instanz des Dynpros existiert, d.h. noch im Dynprostapel vorhanden ist (z.B. call screen 100, call transaction)</p> |
| no_autodef_progid_dynnr | automatische Definition der Programm-ID und der Dynpronummer ausschalten | <p>space Programm-ID und Dynpronummer werden automatisch gezogen (Defaultwert)</p> <p>'x' automatische Definition der Programm-ID und der Dynpronummer ausgeschaltet</p> |

SAP Dialogbox Container

Verwendung

Mit dem SAP Dialogbox Container stellen Sie Controls in einem frei beweglichen, amodalen Dialogfenster dar. Als weitere Möglichkeit können Sie Controls in einen Fullscreen anzeigen.

Im Gegensatz zum [SAP Docking Container \[Seite 26\]](#) können Sie das amodale Dialogfenster nicht an ein Dynpro anbinden.

Methoden

[CONSTRUCTOR \[Seite 20\]](#)

[SET_CAPTION \[Seite 22\]](#)

CONSTRUCTOR

CONSTRUCTOR

Verwendung

Der Konstruktor kreiert, initialisiert und bindet das Control.



Der Konstruktor wird automatisch bei der Erzeugung einer Instanz (`create object`) aufgerufen. Die Parameter übergeben Sie normalerweise bei diesem Aufruf.

Funktionsumfang

create object dialogbox_container

```
exporting
  parent = parent
  width = width
  height = height
  style = style
  dynnr = dynnr
  repid = repid
  lifetime = lifetime
  top = top
  left = left
  caption = caption
  no_autodef_progid_dynnr = no_autodef_progid_dynnr
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2
  create_error = 3
  lifetime_error = 4
  lifetime_dynpro_dynpro_link = 5.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|---|
| parent | Parent der Instanz; d.h., Container in dem das Control angezeigt wird | |
| width | Breite der Dialogbox | |
| height | Höhe der Dialogbox | |
| style | Steuerung des Erscheinungsbild und des Verhaltens des Controls | Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit WS_*) |
| dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| repid | Report-ID: Report, an den der Container gebunden werden soll | |

CONSTRUCTOR

| | | |
|-------------------------|--|---|
| lifetime | Parameter des Lifetime Managements (Lebensdauer des Controls) | <p>cntl_lifetime_imode: Das Control lebt solange der interne Modus nicht abgebaut ist (z.B. <code>leave program</code>, <code>leave transaction</code>)</p> <p>cntl_lifetime_dynpro: Das Control lebt solange die Instanz des Dynpros existiert, d.h. noch im Dynprostapel vorhanden ist (z.B. <code>call screen 100</code>, <code>call transaction</code>)</p> |
| top | Erscheinungsposition der Dialogbox: oberer Rand | |
| left | Erscheinungsposition der Dialogbox: linker Rand | |
| caption | Dialogboxüberschrift | |
| no_autodef_progid_dynnr | automatische Definition der Programm-ID und der Dynpronummer ausschalten | <p>space Programm-ID und Dynpronummer werden automatisch gezogen (Defaultwert)</p> <p>·x· automatische Definition der Programm-ID und der Dynpronummer ausgeschaltet</p> |

SET_CAPTION

SET_CAPTION

Verwendung

Mit dieser Methoden können Sie die Überschrift des Dialogfensters neu setzen.

Funktionsumfang

call method dialogbox_container->set_caption

exporting
caption = caption

exceptions

cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|----------------|
| caption | Überschrift des amodalen Dialogfensters | |

Ereignisse

[CLOSE \[Seite 24\]](#)

CLOSE

CLOSE

Verwendung

Mit diesem Ereignis können Sie auf das Auslösen des Close-Buttons auf der Dialogbox reagieren. Falls Sie das Ereignis entsprechen verarbeiten, können Sie nach dem Auslösen die Dialogbox schließen.

Weitere Informationen finden Sie unter [Ereignisbehandlung \[Extern\]](#) und [Lektion 2: Ereignisbehandlung \[Extern\]](#) des Controls Tutorial.

Funktionsumfang

Folgendes Coding müssen Sie in Ihr Programm einfügen.

Deklarationsteil

```
data close_event type cntl_simple_events.
```

```
class lcl_dialogbox_handler definition.
```

```
public section.
```

```
methods: on_dialogbox_close
```

```
for event close of cl_gui_dialogbox_container
```

```
importing sender.
```

```
endclass.
```

```
class lcl_dialogbox_handler implementation.
```

```
method on_dialogbox_close.
```

```
if not sender is initial.
```

```
call method sender->free
```

```
exceptions
```

```
others = 1.
```

```
if sy-subrc <> 0.
```

```
*Fehlerbehandlung
```

```
endif.
```

```
free dialogbox_container.
```

```
clear dialogbox_container.
```

```
endif.
```

```
endmethod.
```

```
endclass.
```

data: dialogbox_handler type ref to lcl_dialogbox_handler.

Nach dem Erzeugen des Dialogbox Containers

If dialogbox_handler is initial.

 create object dialogbox_handler.

endif.

set handler dialogbox_handler->on_dialogbox_close

for dialogbox_container.

SAP Docking Container

Verwendung

Mit dem SAP Docking Container binden Sie einen oder mehrere Bereiche an ein Dynpro. Sie können den Bereich an alle vier Ränder (links, rechts, oben, unten) des Dynpros anbinden. Das Dynpro wird dabei zugunsten des Docking Containers verkleinert. Der angebundene Bereich kann vom Dynpro gelöst und wieder integriert werden (floating).

Das Verhalten der Bereiche wird über die Reihenfolge der Initialisierung geregelt. Das Anbinden der Docking Container erfolgt von innen nach außen, d.h., beim nächsten Kreieren eines Containers wird dieser an den Rand des Dynpros gebunden und der vorherige Container nach außen geschoben. Sie können technisch gesehen eine beliebige Anzahl von Docking Container an ein Dynpro anbinden, wobei natürlich die Übersichtlichkeit beachtet werden sollte.

Methoden

[CONSTRUCTOR \[Seite 28\]](#)

[DOCK_AT \[Seite 31\]](#)

[FLOAT \[Seite 32\]](#)

[GET_EXTENSION \[Seite 33\]](#)

[SET_EXTENSION \[Seite 35\]](#)


[SET_CAPTION \[Seite 34\]](#)

CONSTRUCTOR

CONSTRUCTOR

Verwendung

Der Konstruktor kreiert, initialisiert und bindet das Control.

 Der Konstruktor wird automatisch bei der Erzeugung einer Instanz (`create object`) aufgerufen. Die Parameter übergeben Sie normalerweise bei diesem Aufruf.

Funktionsumfang

create object docking_container

```
exporting
  parent = parent
  dynnr = dynnr
  repid = repid
  side = side
  extension = extension
  style = style
  lifetime = lifetime
  metric = metric
  caption = caption
  no_autodef_progid_dynnr = no_autodef_progid_dynnr

exceptions
  cntl_error = 1
  cntl_system_error = 2
  create_error = 3
  lifetime_error = 4
  lifetime_dynpro_dynpro_link = 5.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|----------------|
| parent | Parent der Instanz; d.h., Container in dem das Control angezeigt wird | |
| dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| repid | Report-ID: Report, an den der Container gebunden werden soll | |

CONSTRUCTOR

| | | |
|-----------|---|--|
| side | Dynproseite, an die der Bereich angebunden wird | docking_container->dock_at_left: linker Rand docking_container->dock_at_right: rechter Rand docking_container->dock_at_top: oberer Rand docking_container->dock_at_bottom: unterer Rand |
| extension | Extension des Docking Containers, d.h., freier Längenparameter beim Anbinden des Containers (z.B. Docking Container links angebunden, Breite des Docking Containers bestimmbar) | |
| style | Steuerung des Erscheinungsbild und des Verhaltens des Controls | Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit WS_*) |
| lifetime | Parameter des Lifetime Managements (Lebensdauer des Controls) | cntl_lifetime_imode: Das Control lebt solange der interne Modus nicht abgebaut ist (z.B. leave program , leave transaction) cntl_lifetime_dynpro: Das Control lebt solange die Instanz des Dynpros existiert, d.h. noch im Dynprostapel vorhanden ist (z.B. call screen 100 , call transaction) |
| metric | Metrik | docking_container-> metric_default: Dynprometrik (Default) docking_container-> metric_pixel: Pixel docking_container-> metric_mm: Millimeter |
| caption | Überschrift | beliebiges Charakterfeld |

CONSTRUCTOR

| | | |
|-------------------------|--|---|
| no_autodef_progid_dynnr | automatische Definition der Programm-ID und der Dynpronummer ausschalten | space Programm-ID und Dynpronummer werden automatisch gezogen (Defaultwert) 'x' automatische Definition der Programm-ID und der Dynpronummer ausgeschaltet |
|-------------------------|--|---|

DOCK_AT

Verwendung

Mit dieser Methode binden Sie den Docking Container an einen der vier Ränder (links, rechts, oben, unten) des Dynpros um ("umdocken" des Containers).



Sie können das Anbinden an das Dynpro bereits bei der Instanziierung ([CONSTRUCTOR \[Seite 28\]](#)) festlegen.

Funktionsumfang

call method docking_container->dock_at

exporting
side = side

exceptions
cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|--|
| side | Rand des Dynpros, an den der Container angebunden werden soll | docking_container->dock_at_left: linker Rand docking_container->dock_at_right: rechter Rand docking_container->dock_at_top: oberer Rand docking_container->dock_at_bottom: unterer Rand |

FLOAT

FLOAT

Zur Zeit ist diese Methode noch nicht aktiv.

Verwendung

Diese Methode wechselt den Status des Docking Containers von gebunden (docking) auf frei beweglich (floating) und umgekehrt.

Funktionsumfang

call method docking_container->float

```
exporting
  do_float = do_float
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|-------------------------------|--|
| do_float | Status des Docking Containers | docking_container->property_docking : Docking Container ist gebunden (docking) docking_container->property_floating : Docking Container ist frei beweglich (floating) |

GET_EXTENSION

Verwendung

Mit dieser Methoden erhalten Sie den aktuellen Wert der Extension (weitere Informationen finden Sie unter [CONSTRUCTOR \[Seite 28\]](#)).

Funktionsumfang

call method docking_container->get_extension

```
importing
    extension = extension
```

```
exceptions
    cntl_error = 1
    cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|----------------|
| extension | Extension des Docking Containers, d.h., freier Längenparameter beim Anbinden des Containers | |

SET_CAPTION

SET_CAPTION

Verwendung

Mit dieser Methoden können Sie die Überschrift des Docking Containers neu setzen.

Funktionsumfang

call method docking_container->set_caption

exporting
caption = caption

exceptions

cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------------------------|----------------|
| caption | Überschrift des Docking Containers | |

SET_EXTENSION

Verwendung

Mit dieser Methoden setzen Sie den Wert der Extension (weitere Informationen finden Sie unter [GET_EXTENSION \[Seite 33\]](#) und [CONSTRUCTOR \[Seite 28\]](#)).

Funktionsumfang

call method docking_container->set_extension

```
exporting
  extension = extension
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---|----------------|
| extension | Extension des Docking Containers, d.h., freier Längenparameter beim Anbinden des Containers | |

Statische Konstanten

Statische Konstanten

| Statische Konstante | Type / Value | Beschreibung / Verwendung |
|----------------------------|---------------------|----------------------------------|
| dock_at_left | Type Value 1 | linker Rand des Dynpros |
| dock_at_top | Type Value 2 | oberer Rand des Dynpros |
| dock_at_bottom | Type Value 4 | unterer Rand des Dynpros |
| dock_at_right | Type Value 8 | rechter Rand des Dynpros |
| property_docking | Type Value 470 | Docking-Eigenschaft |
| property_floating | Type Value 480 | Floating-Eigenschaft |

SAP Splitter Container

Verwendung

Mit dem SAP Splitter Container stellen Sie mehrere Controls in Zellen eines Sektors dar und gruppieren diese in ihm. Der Splitter Container verwaltet diese Zellen und regelt die Visualisierung dieser. Die Größe der Zellen ist durch eine Split Bar veränderbar und die Zellen sind mit der Split Bar gegeneinander verschiebbar.

Den Splitter Container können Sie nur in einem anderen Container (Parent) anzeigen, z.B. in einem Custom Container oder Docking Container. Durch Einbinden eines Splitter Containers in eine Zelle eines anderen Splitter Containers können Sie Controls schachteln.

Das Grid des Splitters ist initial auf 0x0 eingestellt, die maximale Unterteilung beträgt 16x16. Die Größe der Zeilen und Spalten stellen Sie entweder auf Pixel (absolut) oder Prozent (relativ), Defaulteinstellung relativ. Die Split Bar ist auf nicht beweglich einstellbar.

Falls Sie einen Sektor in nur zwei Zellen (horizontal oder vertikal) aufteilen möchten, verwenden Sie den [SAP Easy Splitter Container \[Seite 62\]](#).

Methoden**Methoden**

[ADD_CONTROL \[Seite 39\]](#)

[ADD_CONTROL_HANDLE \[Seite 40\]](#)

[CONSTRUCTOR \[Seite 41\]](#)

[GET_CONTAINER \[Seite 43\]](#)

[GET_COLUMNS \[Seite 44\]](#)

[GET_COLUMN_MODE \[Seite 45\]](#)

[GET_COLUMN_WIDTH \[Seite 46\]](#)

[GET_COLUMN_SASH \[Seite 47\]](#)

[GET_ROWS \[Seite 48\]](#)

[GET_ROW_HEIGHT \[Seite 49\]](#)

[GET_ROW_MODE \[Seite 50\]](#)

[GET_ROW_SASH \[Seite 51\]](#)

[REMOVE_CONROL \[Seite 52\]](#)

[SET_BORDER \[Seite 53\]](#)

[SET_COLUMN_MODE \[Seite 54\]](#)

[SET_COLUMN_WIDTH \[Seite 55\]](#)

[SET_COLUMN_SASH \[Seite 56\]](#)

[SET_GRID \[Seite 57\]](#)

[SET_ROW_HEIGHT \[Seite 58\]](#)


[SET_ROW_MODE \[Seite 59\]](#)

[SET_ROW_SASH \[Seite 60\]](#)

ADD_CONTROL

Verwendung

Diese Methode fügt ein Control in die gewählte Zelle ein. Falls sich ein bereits existierendes Control in der Zelle befindet, wird dieses nicht gelöscht. Die Sichtbarkeit steuern Sie mit der Methode [SET_VISIBLE \[Seite 88\]](#).

 Das einzufügende Control muß bereits erzeugt sein und auf dem entsprechenden Pop-up Level existieren (Abfrage mit der Methode [IS_ALIVE \[Seite 83\]](#)).

Funktionsumfang

call method splitter_container->add_control

exporting
 row = row
 column = column
 control = control

importing
 result = result

exceptions
 cntl_error = 1
 cntl_system_error = 2
 lifetime_error = 3.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|--|----------------|
| row | Zeile, in die das Control eingefügt werden soll | |
| column | Spalte, in die das Control eingefügt werden soll | |
| control | Instanz des einzufügenden Controls | |
| result | Ergebniscode | |

ADD_CONTROL_HANDLE

ADD_CONTROL_HANDLE




Diese Methode sollten Sie nicht benutzen. Bitte verwenden Sie die Methode [ADD_CONTROL \[Seite 39\]](#).

CONSTRUCTOR

Verwendung

Der Konstruktor kreiert, initialisiert und bindet das Control.

 Der Konstruktor wird automatisch bei der Erzeugung einer Instanz (**create object**) aufgerufen. Die Parameter übergeben Sie normalerweise bei diesem Aufruf.

Funktionsumfang

create object splitter_container

```
exporting
  link_dynnr = link_dynnr
  link_repid = link_repid
  shellstyle = shellstyle
  left = left
  top = top
  width = width
  height = height
  metric = metric
  align = align
  parent = parent
  rows = rows
  columns = columns
  no_autodef_progid_dynnr = no_autodef_progid_dynnr

exceptions
  cntl_error = 1
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|------------|--|---|
| link_dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| link_repid | Report-ID: Report, an den der Container gebunden werden soll | |
| shellstyle | Steuerung des Erscheinungsbild und des Verhaltens des Controls | Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit WS_*) |
| left | linke Position des Containers | |
| top | obere Position des Containers | |
| width | Breite des Containers | |
| height | Höhe des Containers | |

CONSTRUCTOR

| | | |
|-------------------------|---|---|
| metric | Metrik | <code>splitter_container->metric_default:</code> Dynprometrik (Default) <code>splitter_container->metric_pixel:</code> Pixel <code>splitter_container->metric_mm:</code> Millimeter |
| align | Ausrichtung des Splitter Containers | Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit ALIGN_*) |
| parent | Parent der Instanz, d.h., Container an den das Control gebunden werden soll | |
| rows | Anzahl der Zeilen | maximal 16 Zeilen |
| columns | Anzahl der Spalten | maximal 16 Spalten |
| no_autodef_progid_dynnr | automatische Definition der Programm-ID und der Dynpronummer ausschalten | space Programm-ID und Dynpronummer werden automatisch gezogen (Defaultwert) 'x' automatische Definition der Programm-ID und der Dynpronummer ausgeschaltet |

GET_CONTAINER

Verwendung

Diese Methode liefert die Referenzvariable für einen ausgewählte Zelle. Diese Referenzvariable geben Sie beim Binden eines neuen Controls an.

Funktionsumfang

call method splitter_container->get_container

exporting
row = row
column = column

receiving
container = container

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------|----------------|
| row | Zeile der Zelle | |
| column | Spalte der Zelle | |

GET_COLUMNS**GET_COLUMNS****Verwendung**

Diese Methode liefert die Anzahl der Spalten des Splitter Containers.

Funktionsumfang

call method splitter_container->get_columns

```
importing  
  result = result
```

```
exceptions  
  cntl_error = 1  
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|-----------|----------------|
| result | Ergebnis | |

GET_COLUMN_MODE

Verwendung

Diese Methode liefert den Modus (absolut oder relativ) der Spalten des Splitter Containers.
Weitere Informationen finden Sie unter [SAP Splitter Container \[Seite 37\]](#).

Funktionsumfang

call method splitter_container->get_column_mode

```
importing  
  result = result
```

```
exceptions  
  cntl_error = 1  
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|-----------|----------------|
| result | Ergebnis | |

GET_COLUMN_WIDTH

GET_COLUMN_WIDTH

Verwendung

Diese Methode liefert die Breite einer Spalte.

Funktionsumfang

call method splitter_container->get_column_width

exporting

id = id

importing

result = result

exceptions

cntl_error = 1

cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|---------------|----------------|
| id | ID der Spalte | |
| result | Ergebnis | |

GET_COLUMN_SASH

Verwendung

Diese Methode liefert die Konfiguration der Spalten der Split Bar.

Funktionsumfang

call method splitter_container->get_column_sash

exporting
 id = id
 type = type

importing
 result = result

exceptions
 cntl_error = 1
 cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------------|--|
| id | ID der Split Bar | Die erste Split Bar hat den Wert 1, usw. |
| type | Attribut der Split Bar | type_movable |
| result | Ergebnis | <code>splitter_container->>true</code> : bewegliche Split Bar <code>splitter_container->>false</code> : feste Split Bar |

GET_ROWS**GET_ROWS****Verwendung**

Diese Methode liefert die Anzahl der Zeilen des Splitter Containers.

Funktionsumfang

call method splitter_container->get_rows

```
importing  
  result = result
```

```
exceptions  
  cntl_error = 1  
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|-----------|----------------|
| result | Ergebnis | |

GET_ROW_HEIGHT

Verwendung

Diese Methode liefert die Höhe einer Zeile.

Funktionsumfang

call method splitter_container->get_row_height

exporting

id = id

importing

result = result

exceptions

cntl_error = 1

cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|--------------|----------------|
| id | ID der Zeile | |
| result | Ergebnis | |

GET_ROW_MODE

GET_ROW_MODE

Verwendung

Diese Methode liefert den Modus (absolut oder relativ) der Zeilen des Splitter Containers. Weitere Informationen finden Sie unter [SAP Splitter Container \[Seite 37\]](#).

Funktionsumfang

call method splitter_container->get_row_mode

```
importing  
  result = result
```

```
exceptions  
  cntl_error = 1  
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|-----------|----------------|
| result | Ergebnis | |

GET_ROW_SASH

Verwendung

Diese Methode liefert die Konfiguration der Zeilen der Split Bar.

Funktionsumfang

call method splitter_container->get_row_sash

exporting
 id = id
 type = type

importing
 result = result

exceptions
 cntl_error = 1
 cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------|--|
| id | ID der Split Bar | Die erste Split Bar hat den Wert 1, usw. |
| type | Attribut | type_movable |
| result | Ergebnis | splitter_container->>true : bewegliche Split Bar splitter_container->>false : feste Split Bar |

REMOVE_CONTROL

REMOVE_CONTROL

Verwendung

Diese Methode entfernt ein ausgewähltes Control aus dem Splitter Container.



Das Control wird nicht abgebaut. Falls Sie es abbauen möchten, müssen Sie die Methode [FREE \[Seite 90\]](#) aufrufen und gegebenenfalls die Referenzvariable abbauen (Weitere Informationen finden Sie unter [Lifetime Management \[Extern\]](#)).

Funktionsumfang

call method splitter_container->remove_control

exporting
row = row
column = column

importing
result = result

exceptions
cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------|----------------|
| row | Zeile der Zelle | |
| column | Spalte der Zelle | |
| result | Ergebnis | |

SET_BORDER

Verwendung

Diese Methode schaltet den Rahmen um den Splitter Container an oder aus.



Das Ausschalten des Rahmens benutzen Sie beim Schachteln von Splitter Containern.

Funktionsumfang

call method splitter_container->set_border

exporting
border = border

exceptions
cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|----------------------|---|
| border | Zeichnen des Rahmens | cl_gui_cfw=>true: Rahmen zeichnen cl_gui_cfw=>false: Rahmen nicht zeichnen |

SET_COLUMN_MODE

SET_COLUMN_MODE

Verwendung

Diese Methode setzt den Modus der Spalten des Splitter Containers auf absolut oder relativ. Weitere Informationen finden Sie unter [SAP Splitter Container \[Seite 37\]](#).

Funktionsumfang

call method splitter_container->set_column_mode

exporting
mode = mode

importing
result = result

exceptions
cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------|---|
| mode | Modus der Spalte | <code>splitter_container->mode_absolut</code> : absoluter Modus (Pixel) <code>splitter_container->mode_relative</code> : relativer Modus (Prozent) |
| result | Ergebnis | |

SET_COLUMN_WIDTH

Verwendung

Diese Methode setzt Breite einer Spalte.

Funktionsumfang

call method splitter_container->set_column_width

exporting
 id = id
 width = width

importing
 result = result

exceptions
 cntl_error = 1
 cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|-------------------|----------------|
| id | ID der Spalte | |
| width | Breite der Spalte | |
| result | Ergebnis | |

SET_COLUMN_SASH

SET_COLUMN_SASH

Verwendung

Diese Methode setzt die Konfiguration der Split Bar der Spalten.

Funktionsumfang


call method splitter_container->set_column_sash

```
exporting
  id = id
  type = type
  value = value
```

```
importing
  result = result
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------------|--|
| id | ID der Split Bar | Die erste Split Bar hat den Wert 1, usw. |
| type | Attribut der Split Bar | type_movable |
| value | Wert des Attributs | <code>splitter_container->>true</code> : bewegliche Split Bar <code>splitter_container->>false</code> : feste Split Bar |
| result | Ergebnis | |

 Falls Sie das Attribut der Split Bar ändern möchten, wählen Sie für den Parameter `type` den Wert `type_movable` und dann den entsprechend Wert des Attributs.

SET_GRID

Verwendung

Diese Methode erzeugt ein neues Gitternetz (Grid) der Zellen. Das bereits existierende wird abgebaut. Das Gitternetz übergeben Sie bereits beim Erzeugen des Splitter Containers (Weitere Informationen finden Sie unter [CONSTRUCTOR \[Seite 41\]](#)).



Diese Methode sollten Sie zur Zeit aufgrund von unerwarteten Ergebnisse am Frontend noch nicht benutzen.

Funktionsumfang

call method splitter_container->set_grid

exporting

rows = rows

columns = columns

importing

result = result

exceptions

cntl_error = 1

cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|--------------------|----------------|
| rows | Anzahl der Spalten | |
| columns | Anzahl der Zeilen | |
| result | Ergebnis | |

SET_ROW_HEIGHT

SET_ROW_HEIGHT

Verwendung

Diese Methode setzt die Höhe einer Zeile.

Funktionsumfang

call method splitter_container->set_row_height

exporting
id = id
height = height

importing
result = result

exceptions
cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|----------------|----------------|
| id | ID der Zeile | |
| height | Höhe der Zeile | |
| result | Ergebnis | |

SET_ROW_MODE

Verwendung

Diese Methode setzt den Modus der Zeilen des Splitter Containers auf absolut oder relativ. Weitere Informationen finden Sie unter [SAP Splitter Container \[Seite 37\]](#).

Funktionsumfang

call method splitter_container->set_row_mode

exporting
mode = mode

importing
result = result

exceptions
cntl_error = 1
cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------|---|
| mode | Modus der Zeilen | <code>splitter_container_mode_absolut</code> : absoluter Modus (Pixel) <code>splitter_container_mode_relative</code> : relativer Modus (Prozent) |
| result | Ergebnis | |

SET_ROW_SASH

SET_ROW_SASH

Verwendung

Diese Methode setzt die Konfiguration der Split Bar der Zeilen.

Funktionsumfang


call method splitter_container->set_row_sash

```
exporting
  id = id
  type = type
  value = value
```

```
importing
  result = result
```

```
exceptions
  cntl_error = 1
  cntl_system_error = 2.
```

| Parameter | Bedeutung | Mögliche Werte |
|-----------|------------------------|--|
| id | ID der Split Bar | Die erste Split Bar hat den Wert 1, usw. |
| type | Attribut der Split Bar | type_movable |
| value | Wert des Attributs | <code>splitter_container->>true</code> : bewegliche Split Bar <code>splitter_container->>false</code> : feste Split Bar |
| result | Ergebnis | |

 Falls Sie das Attribut der Split Bar ändern möchten, wählen Sie für den Parameter `type` den Wert `type_movable` und dann den entsprechend Wert des Attributs.

Statische Konstanten

| Statische Konstante | Type / Value | Beschreibung / Verwendung |
|---------------------|----------------|----------------------------------|
| mode_absolute | Type Value 0 | absoluter Modus (Pixel) |
| mode_relative | Type Value 1 | relativer Modus (Prozent) |
| type_movable | Type Value 0 | Wert für verschiebbare Split Bar |
| true | Type Value 1 | Attributwert: wahr |
| false | Type Value 0 | Attributwert falsch |

SAP Easy Splitter Container

Verwendung

Der SAP Easy Splitter Container ist eine Version des SAP Splitter Containers mit eingeschränkter Funktionalität. Mit dem Easy Splitter Container stellen Sie Controls in zwei Zellen eines Sektors dar. Sie können die Split Bar zwischen den zwei Zellen entweder horizontal oder vertikal wählen.

Den Easy Splitter Container können Sie nur in einem anderen Container (Parent) anzeigen, z.B. in einem Custom Container oder Docking Container.

Der Easy Splitter Container ist beliebig oft ineinander schachtelbar.

Falls Sie weitergehende Funktionalitäten benötigen, verwenden Sie den [SAP Splitter Container \[Seite 37\]](#).

Methoden

[CONSTRUCTOR \[Seite 64\]](#)

[SET_SASH_POSITION \[Seite 66\]](#)

CONSTRUCTOR

CONSTRUCTOR

Verwendung

Der Konstruktor kreiert, initialisiert und bindet das Control.



Der Konstruktor wird automatisch bei der Erzeugung einer Instanz (`create object`) aufgerufen. Die Parameter übergeben Sie normalerweise bei diesem Aufruf.

Funktionsumfang

create object easy_splitter_container

exporting

link_dynnr = link_dynnr

link_repid = link_repid

metric = metric

parent = parent

orientation = orientation

sash_position = sash_position

with_border = with_border

exceptions

cntl_error = 1

cntl_system_error = 2.

| Parameter | Bedeutung | Mögliche Werte |
|-------------|---|--|
| link_dynnr | Nummer des Dynpros, an dem das Control gebunden werden soll | |
| link_repid | Report-ID: Report, an den der Container gebunden werden soll | |
| metric | Metrik | <code>easy_splitter_container->metric_default</code> : Dynprometrik (Default) <code>easy_splitter_container->metric_pixel</code> : Pixel <code>easy_splitter_container->metric_mm</code> : Millimeter |
| parent | Parent der Instanz; d.h., Container in dem das Control angezeigt wird | |
| orientation | Orientierung der Split Bar | <code>easy_splitter_container->orientation_vertical</code> : horizontale Split Bar <code>easy_splitter_container->orientation_horizontal</code> : vertikale Split Bar |

CONSTRUCTOR

| | | |
|---------------|-------------------------|--|
| sash_position | Position der Split Bar | prozentuale Werte; von links nach rechts oder von oben nach unten Default: 50 |
| with_border | Einstellung des Rahmens | 0: ohne Rahmen 1: mit Rahmen |

SET_SASH_POSITION

SET_SASH_POSITION

Verwendung

Mit dieser Methode können Sie die Position der Split Bar neu festlegen.

Funktionsumfang

call method easy_splitter_container->set_sash_position

exporting

sash_position = sash_position

| Parameter | Bedeutung | Mögliche Werte |
|---------------|------------------------|---|
| sash_position | Position der Split Bar | prozentuale Werte; von links nach rechts oder von oben nach unten |

Statische Attribute

| Statisches Attribut | Type / Value | Beschreibung / Verwendung |
|------------------------|----------------|--|
| orientation_horizontal | Type Value 1 | horizontale Orientierung der Split Bar |
| orientation_vertical | Type Value 0 | vertikale Orientierung der Split Bar |

Instanz-Attribute

Instanz-Attribute

| Instanz-Attribut | Type / Value | Beschreibung / Verwendung |
|-------------------------|---------------------------------|---|
| top_left_container | Type ref to CL_GUI_CONTAINER | ID des linken oder oberen Containers |
| bottom_right_container | Type ref to CL_GUI_CONTAINER | ID des rechten oder unteren Containers |
| pane_orientation | Type I | Orientierung der Unterbereiche, bzw. der Split Bar |

Methoden der Klasse CL_GUI_CFW

Die Klasse `CL_GUI_CFW` beinhaltet statische Methoden, die beim Aufruf auf **alle** instanziierten Custom Controls wirken.

dispatch

dispatch

Die Methode `dispatch` verteilt Applikationsereignisse ([siehe Ereignisbehandlung \[Extern\]](#)) an die für das Ereignis angemeldeten Ereignisbehandler. Wenn diese Methode nicht im Applikationsprogramm innerhalb von PAI aufgerufen wird, dann wird sie automatisch vom System nach dem Abarbeiten von PAI aufgerufen. Die Methode liefert einen Returncode zurück, über den der Erfolg des Aufrufs abzulesen ist.

```
CALL METHOD cl_gui_cfw=>dispatch
      IMPORTING return_code = return_code.
```

| Parameter | Bedeutung |
|--------------------------|---|
| <code>return_code</code> | <p><code>cl_gui_cfw=>rc_found</code>: Das Ereignis konnte erfolgreich an eine Behandlungsmethode übergeben werden.</p> <p><code>cl_gui_cfw=>rc_unknown</code>: Das Ereignis wurde nicht in der Ereignisliste registriert.</p> <p><code>cl_gui_cfw=>rc_noevent</code>: Es wurde kein Ereignis auf einem Control ausgelöst. Der <code>OK_CODE</code> war daher ein normaler <code>OK_CODE</code> (z.B. von einem Menüeintrag).</p> <p><code>cl_gui_cfw=>rc_nodispatch</code>: Dem Ereignis konnte keine Behandlungsmethode zugeordnet werden.</p> |



Das Ereignis kann nur einmalig verteilt werden. Danach ist das Ereignis verbraucht. Daher wird ein zweiter Aufruf der Methode nicht nochmals zu einem Sprung in den Ereignisbehandler führen.

flush

Mit dieser Methode synchronisieren Sie explizit die [Automation Queue \[Extern\]](#). Die gepufferten Operationen werden dann zum Frontend per GUI-RFC geschickt. Dort wird die Automation Queue in der Reihenfolge abgearbeitet, wie Sie sie gefüllt haben.

Im Fehlerfall wird eine Ausnahme ausgelöst, die Sie auf jeden Fall abfragen und behandeln sollten. Da eine Zuordnung des Fehlers in der Regel nicht mehr möglich ist, stehen Ihnen sowohl im Debugger als auch im SAP GUI Werkzeuge zur Verfügung, um den Fehler zu lokalisieren:

Debugger: Markieren Sie in den Einstellungen das *Ankreuzfeld Automation Controller: Aufträge immer synchron verarbeiten*. Dies führt dazu, daß nach jeder Methode, die den Automation Controller ruft, die Methode `c1_gui_cfw=>flush` automatisch aufgerufen wird.

SAP GUI: In den Einstellungen zum SAP GUI können Sie auf der Karteikarte *Trace* das Ankreuzfeld *Automation* wählen. Dadurch wird die Kommunikation zwischen Applikationsserver und Automation Controller in einer Trace-Datei mitgeschrieben. Diese kann dann ausgewertet werden.

```
CALL METHOD c1_gui_cfw=>flush
          EXCEPTIONS CNTL_SYSTEM_ERROR = 1
                   CNTL_ERROR = 2.
```



Führen Sie nur so viele Synchronisationspunkte in Ihr Programm ein, wie wirklich nötig sind. Bei jeder Synchronisation wird nämlich eine RFC-Verbindung zum SAP GUI geöffnet.

get_living_dynpro_controls**get_living_dynpro_controls**

Mit dieser Methode können Sie sich eine Liste von Referenzvariablen zu allen noch aktiven Custom Controls besorgen.

```
CALL METHOD cl_gui_cfw=>get_living_dynpro_controls  
IMPORTING control_list = control_list.
```

| Parameter | Bedeutung |
|---------------------------|---|
| <code>control_list</code> | Liste der Referenzvariablen zu aktiven Custom Controls. Liste ist vom Typ <code>CNTO_CONTROL_LIST</code> (in der Klasse <code>CL_GUI_CFW</code> definiert) |

set_new_ok_code

Diese Methode darf nur in Behandlermethoden zu Systemereignissen eingesetzt werden. Sie setzt einen `OK_CODE`, der ein Ausführen von `PAI` nach sich zieht. Dadurch können Sie nach dem Feldtransport nochmals die Kontrolle in Ihren `PAI`-Modulen bekommen.

```
CALL METHOD cl_gui_cfw=>set_new_ok_code
      EXPORTING new_code = new_code
      IMPORTING   rc = rc.
```

| Parameter | Bedeutung |
|--------------------------|--|
| <code>new_code</code> | Funktionscode, der in das <code>OK_CODE</code> -Feld (<code>SY-UCOMM</code>) gestellt werden soll. |
| <code>return_code</code> | <p><code>cl_gui_cfw=>rc_posted</code>: Der <code>OK_CODE</code> wurde mit Erfolg gesetzt, und die Verarbeitung wird nach Abschluß der Behandlermethode mit <code>PAI</code> fortgesetzt (vorher wird noch die automatische Feldprüfung des Dynpros durchgeführt).</p> <p><code>cl_gui_cfw=>rc_wrong_state</code>: Die Methode wurde nicht bei einem Systemereignis aufgerufen.</p> <p><code>cl_gui_cfw=>rc_invalid</code>: Der gesetzte <code>OK_CODE</code> ist kein erlaubter <code>OK_CODE</code>.</p> |

update_view

update_view

Die Automation Queue wird durch den Aufruf der Methode [flush \[Seite 71\]](#) nur dann synchronisiert, wenn in der Automation Queue Returnwerte enthalten sind.

Für alle Fälle, in denen auch im Fall einer Returnwert-freien Queue gewünscht wird, daß die Automation Queue synchron versendet wird, gibt es im Control Framework die Methode `CL_GUI_CFW=>UPDATE_VIEW`. Diese Methode darf nur dann verwendet werden, wenn es zwingend notwendig ist, ein Update des SAP GUI zu erreichen. Beispiele hierfür sind sehr lange laufende Anwendungen, die in regelmäßigen Abständen dem Benutzer ein Feedback über den Fortschritt der Aktion anzeigen möchten.

```
CALL METHOD cl_gui_cfw=>update_view
          EXCEPTIONS CNTL_SYSTEM_ERROR = 1
                   CNTL_ERROR         = 2.
```

Methoden der Klasse CL_GUI_CONTROL

Die Klasse `CL_GUI_CONTROL` beinhaltet Methoden, die zum Setzen von Control-Eigenschaften (z.B. Visualisieren des Controls), Registrieren von Ereignissen und zum Abbau des Controls dienen.

constructor

constructor

Diese Methode wird von der Control-Verschaltung des verwendeten Controls aufgerufen, um ein Control am Frontend zu instanzieren.



Um ein SAP Control zu instanzieren, rufen Sie immer den Konstruktor der dazugehörigen Klasse auf.

```
CREATE OBJECT my_control
  EXPORTING    clsid          = clsid
              lifetime       = lifetime
              shellstyle     = shellstyle
              parent         = parent
              autoalign      = autoalign
  EXCEPTIONS  cntl_error     = 1
              cntl_system_error = 2
              create_error    = 3
              lifetime_error  = 4.
```

| Parameter | Bedeutung |
|-------------------------|---|
| <code>clsid</code> | ID der Klasse |
| <code>lifetime</code> | <p>Parameter für das Lifetime Management. Folgende Werte sind möglich:</p> <p><code>my_control->lifetime_imode</code>: Das Control lebt, solange der interne Modus nicht abgebaut wird (z.B.: <code>leave program. leave transaction. set screen 0, leave screen.</code>). Danach wird die Methode finalize [Seite 78] aufgerufen.</p> <p><code>my_control->lifetime_dynpro</code>: Das Control lebt, solange die Instanz des Dynpros existiert, d.h. sich noch im Dynprostapel befindet. Danach wird die Methode free [Seite 90] aufgerufen. Die Benutzung dieses Modus regelt automatisch die Sichtbarkeit von Controls. Controls werden immer nur dann eingeblendet, wenn das Dynpro aktiv ist, auf dem sie erzeugt wurden. Ist ein anderes Dynpro aktiv, werden sie automatisch unsichtbar geschaltet.</p> <p><code>my_control->lifetime_default</code>: Wird das Control in einen Container eingebaut, übernimmt es die Lebensdauer des Containers. Wird es nicht in einen Container eingebaut (z.B. weil es selbst ein Container ist), dann wird die Lebensdauer auf <code>my_control->lifetime_imode</code> gesetzt.</p> |
| <code>shellstyle</code> | <p>Steuerung des Erscheinungsbilds und des Verhaltens des Controls</p> <p>Konstanten aus dem ABAP-Include <code><CTLDEF></code>, die mit <code>WS</code> beginnen, können Sie übergeben. Kombinationen von mehreren Styles können Sie durch Addieren der Konstanten erreichen. Der Vorschlagswert führt intern zum Setzen einer ausreichenden Kombination von Style-Konstanten.</p> |
| <code>parent</code> | <p>Container, in dem das SAP Picture Control angezeigt werden kann (siehe SAP Container [Seite 1])</p> |

| | |
|------------------|---|
| autoalign | '': kein automatisches Ausrichten des Controls 'X': automatisches Ausrichten des Controls. Dabei wird der maximal verfügbare Platz innerhalb eines Containers verwendet. |
|------------------|---|

finalize**finalize**

Diese Methode wird von der verwendeten Controlverschalung überdefiniert. In ihr werden control-spezifische Funktionen zum Abbau des Controls aufgerufen. Diese Methode wird automatisch von der Methode [free \[Seite 90\]](#) aufgerufen, bevor das Control am Frontend abgebaut wird.

```
CALL METHOD my_control->finalize.
```

get_focus

Diese statische Methode liefert die Objektreferenz des Custom Controls zurück, welches den Fokus hat.

```
CALL METHOD cl_gui_control=>get_focus
IMPORTING control          = control
EXCEPTIONS cntl_error     = 1
            cntl_system_error = 2.
```

| Parameter | Bedeutung |
|-----------|--|
| control | Objektreferenz (TYPE REF TO cl_gui_control) auf das Control, das den Fokus hat |

get_height**get_height**

Diese Methode liefert die Höhe des Controls zurück.

```
CALL METHOD control->get_height
  IMPORTING height          = height
  EXCEPTIONS cntl_error    = 1.
```

| Parameter | Bedeutung |
|-----------|----------------------------|
| height | Aktuelle Höhe des Controls |

get_registered_events

Diese Methode liefert eine Liste aller für das Custom Control `my_control` registrierten Ereignisse zurück.

```
CALL METHOD my_control->get_registered_events
    IMPORTING events = events
    EXCEPTIONS cntl_error = 1.
```

| Parameter | Bedeutung |
|---------------------|--|
| <code>events</code> | Tabelle der zu registrierenden Ereignisse für das Custom Control <code>my_control</code> |

Die Tabelle `events` ist eine Liste mit Ereignissen, auf die Sie sich registriert haben. Die Tabelle wird mit Bezug auf den Tabellentyp `CNTL_SIMPLE_EVENTS` definiert. Dem Tabellentyp liegt die Struktur `CNTL_SIMPLE_EVENT` zugrunde. Dieser besteht aus folgenden Feldern:

| Feld | Bedeutung |
|-------------------------|--|
| <code>EVENTID</code> | Name des Ereignisses |
| <code>APPL_EVENT</code> | Unterscheidung, ob es sich um ein Systemereignis (initial) oder ein Applikationsereignis (x) handeln soll. |

Die Werte, die dem Feld `EVENTID` zuzuordnen sind, sind control-spezifisch und werden daher bei den entsprechenden Controls beschrieben.



Allgemeine Informationen zur Ereignisbehandlung finden Sie in der Dokumentation des SAP Control Frameworks unter [Ereignisbehandlung \[Extern\]](#).

get_width**get_width**

Diese Methode liefert die Breite des Controls zurück.

```
CALL METHOD control->get_width
    IMPORTING width          = width
    EXCEPTIONS cntl_error    = 1.
```

| Parameter | Bedeutung |
|-----------|------------------------------|
| width | Aktuelle Breite des Controls |

is_alive

Diese Methode liefert als Ergebnis, ob ein Custom Control zu einer Objektreferenz noch am Frontend vorhanden ist.

```
CALL METHOD my_control->is_alive  
RETURNING state = state.
```

| Parameter | Bedeutung |
|-----------|---|
| state | <code>my_control->state_dead</code> : Custom Control ist nicht mehr am Frontend aktiv <code>my_control->state_alive</code> : Custom Control ist auf aktuellem Dynpro aktiv <code>my_control->state_alive_on_other_dynpro</code> : Custom Control ist auf dem aktuellen Dynpro nicht aktiv, aber am Frontend noch aktiv (d.h. unsichtbar) |

set_alignment

set_alignment

Diese Methode richtet das Custom Control innerhalb seines Containers aus:

```
CALL METHOD my_control->set_alignment
    EXPORTING alignment      = alignment
    EXCEPTIONS cntl_error    = 1
               cntl_system_error = 2.
```

| Parameter | Bedeutung |
|-----------|--------------------------|
| alignment | Ausrichtung des Controls |

Der Parameter **alignment** kann aus Kombinationen folgender Ausrichtungen bestehen:

| Name | Bedeutung |
|-----------------------------|-----------------------------|
| my_control->align_at_left | Ausrichtung am linken Rand |
| my_control->align_at_right | Ausrichtung am rechten Rand |
| my_control->align_at_top | Ausrichtung am oberen Rand |
| my_control->align_at_bottom | Ausrichtung am unteren Rand |

Kombinationen erhält man durch Aufaddieren der Komponenten:

alignment = my_control->align_at_left + my_control->align_at_top.

set_focus

Mit dieser statischen Methode können Sie den Fokus auf ein Custom Control setzen.

```
CALL METHOD cl_gui_control=>set_focus
    EXPORTING control          = control
    EXCEPTIONS cntl_error     = 1
               cntl_system_error = 2.
```

| Parameter | Bedeutung |
|-----------|--|
| control | Objektreferenz (TYPE REF TO cl_gui_control) auf das Control, das den Fokus bekommen soll |

set_position**set_position**

Diese Methode plaziert das Control an eine bestimmte Stelle des Dynpros.



In der Regel wird die Position des Controls über seinen Container geregelt.

```
CALL METHOD my_control->set_position
  EXPORTING height      = height
            left        = left
            top         = top
            width       = width
  EXCEPTIONS cntl_error    = 1
            cntl_system_error = 2.
```

| Parameter | Bedeutung |
|-----------|--------------------------|
| height | Höhe des Controls |
| left | Linker Rand des Controls |
| top | Oberer Rand des Controls |
| width | Breite des Controls |

set_registered_events

Mit dieser Methode registrieren Sie sich auf Ereignisse des Controls ([siehe auch: Ereignisbehandlung \[Extern\]](#)).

```
CALL METHOD my_control->set_registered_events
    EXPORTING events          = events
    EXCEPTIONS cntl_error     = 1
               cntl_system_error = 2
               illegal_event_combination = 3.
```

| Parameter | Bedeutung |
|-----------|--|
| events | Tabelle der zu registrierenden Ereignisse für das Custom Control <code>my_control</code> |

Die Tabelle `events` ist eine Liste mit Ereignissen, auf die Sie sich registrieren wollen. Die Tabelle wird mit Bezug auf den Tabellentyp `CNTL_SIMPLE_EVENTS` definiert. Dem Tabellentyp liegt die Struktur `CNTL_SIMPLE_EVENT` zugrunde. Dieser besteht aus folgenden Feldern:

| Feld | Bedeutung |
|------------|--|
| EVENTID | Name des Ereignisses |
| APPL_EVENT | Unterscheidung, ob es sich um ein Systemereignis (initial) oder ein Applikationsereignis (x) handeln soll. |

Die Werte, die dem Feld `EVENTID` zuzuordnen sind, sind control-spezifisch und werden daher bei den entsprechenden Controls beschrieben.

set_visible**set_visible**

Mit dieser Methode können Sie die Sichtbarkeit eines Custom Controls verändern.

```
CALL METHOD my_control->set_visible
    EXPORTING visible          = visible
    EXCEPTIONS cntl_error      = 1
               cntl_system_error = 2.
```

| Parameter | Bedeutung |
|-----------|---|
| visible | x: Custom Control ist sichtbar : Custom Control ist nicht sichtbar |

Methoden der Klasse CL_GUI_OBJECT

Die Klasse `CL_GUI_OBJECT` beinhaltet wichtige Methoden zum Verschalen von Custom Controls. Für Anwendungsprogramme ist einzig die Methode [is_valid \[Seite 91\]](#) relevant.

free

free

Diese Methode baut ein Custom Control am Frontend ab. Nach Aufruf dieser Methode sollten Sie auch die Objektreferenz initialisieren (**FREE my_control**).

```
CALL METHOD my_control->free
      EXCEPTIONS cntl_error      = 1
                 cntl_system_error = 2.
```

is_valid

Diese Methode liefert als Ergebnis, ob ein Custom Control zu einer Objektreferenz noch am Frontend vorhanden ist.

```
CALL METHOD my_control->is_valid  
    IMPORTING result = result.
```

| Parameter | Bedeutung |
|-----------|--|
| result | 0: Custom Control ist nicht mehr am Frontend aktiv 1: Custom Control ist noch aktiv |