

SAP Picture (BC-CI)



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 Picture (BC-CI)	5
Instanz für das SAP Picture	7
Anlegen eines Controls am Beispiel des SAP Picture	8
Arbeiten mit dem SAP Picture	10
Registrieren und Verarbeiten von Ereignissen	12
Ereignisse des SAP Picture	14
Verwendung von Controls im WAN	16
Spezielle Hinweise für das SAP Picture	18
Methoden der Klasse CL_GUI_PICTURE	19
constructor	20
load_picture_from_url	21
clear_picture	22
set_display_mode	23
load_picture_from_sap_icons	24
display_context_menu	25
Methoden des OO Control Frameworks	26
Methoden der Klasse CL_GUI_CFW	27
dispatch	28
flush	29
get_living_dynpro_controls	30
set_new_ok_code	31
update_view	32
Methoden der Klasse CL_GUI_OBJECT	33
is_valid	34
free	35
Methoden der Klasse CL_GUI_CONTROL	36
finalize	37
set_registered_events	38
get_registered_events	39
is_alive	40
set_alignment	41
set_position	42
set_visible	43
get_focus	44
set_focus	45
get_height	46
get_width	47

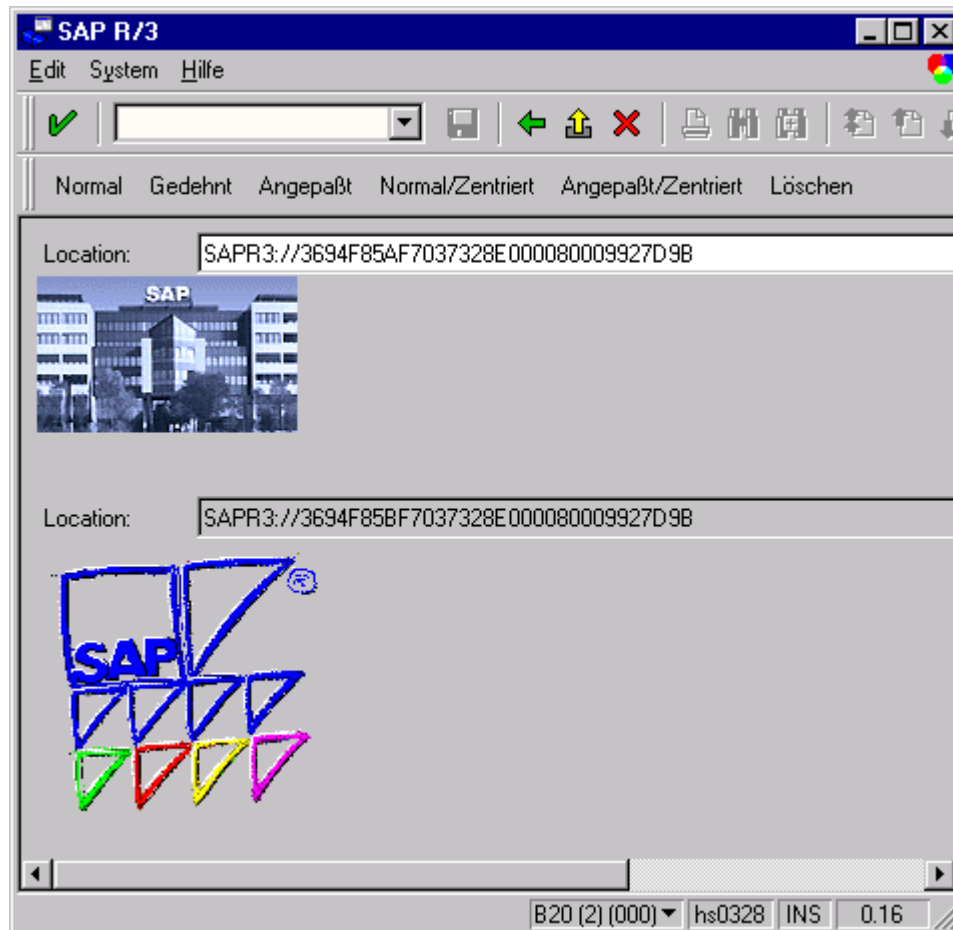
SAP Picture (BC-CI)

Einsatzmöglichkeiten

Das SAP Picture kann zur Anzeige von Grafiken (BMP, JPG und GIF) auf Dynpros benutzt werden.



Das Programm `SAP_PICTURE_DEMO` ist ein Beispielprogramm, in dem zwei Bilder in einem Dynpro eingespielt werden:



Ein weiteres Beispiel zum Anzeigen von SAP-Icons ist das Programm `SAP_PICTURE_DEMO_ICON`.

Voraussetzung

Um mit dem SAP Picture sinnvoll arbeiten zu können, sollten Sie über Grundkenntnisse in [ABAP OO \[Extern\]](#) verfügen. Weiterhin sollten Sie sich mit den Techniken des [SAP Control Frameworks \[Extern\]](#) vertraut gemacht haben.

SAP Picture (BC-CI)**Integration**

Die Grafik wird über eine URL-Adresse identifiziert. Die URL-Adresse kann dabei eine Internet-Adresse (http oder ftp), eine Adresse in einem Verzeichnis (file) oder eine Speicheradresse am Frontend (sapr3 generiert durch den Data Provider) sein.

Funktionsumfang

Das SAP Picture bietet Funktionen zum Anzeigen (Vergrößern, Verkleinern) von Grafiken und SAP-Ikonen an.

Instanz für das SAP Picture

Definition

Diese Instanz wird mit Referenz auf die Klasse `c1_gui_picture` definiert:

```
data picture type ref to c1_gui_picture.
```

Verwendung

Eine Instanz für das SAP Picture verwaltet alle Informationen bezüglich eines Bildes in Ihrem Dynpro. Auf diese Instanz können Sie Methoden aufrufen, mit denen Sie die Eigenschaften des SAP Picture definieren und ändern können.

Integration

Die Klasse `c1_gui_picture` beinhaltet sowohl [control-spezifische Methoden \[Seite 19\]](#) als auch [Methoden des OO Control Frameworks \[Seite 26\]](#).

Anlegen eines Controls am Beispiel des SAP Picture

Anlegen eines Controls am Beispiel des SAP Picture

Voraussetzungen

Der folgende Ablauf ist für alle von SAP ausgelieferten Custom Controls anwendbar. In den Code-Beispielen wird immer auf das SAP Picture Control eingegangen. Für andere Controls ist aber im Prinzip nur die Klasse des Controls auszutauschen.

Weiterhin geht das Beispiel davon aus, daß das Custom Control in einem Custom Container eingebaut wird. Andere Szenarios werden in der Dokumentation zu den Control Containern beschrieben.

Ablauf

Instanz anlegen

1. Definieren Sie eine Referenzvariable für den Custom Container, in dem das Custom Control angezeigt werden soll ([Siehe SAP Container \[Extern\]](#))

```
DATA container TYPE REF TO cl_gui_custom_container.
```

2. Definieren Sie eine Referenzvariable für das Picture Control:

```
DATA picture TYPE REF TO cl_gui_picture.
```

3. Erzeugen Sie den Custom Container. Den Bereich 'CUSTOM' für den Custom Container müssen Sie vorher im Screen Painter angelegt haben. Beim Erzeugen des Containers legen Sie auch seine [Lebensdauer \[Extern\]](#) fest ([siehe constructor \[Extern\]](#)).

```
CREATE OBJECT container  
  EXPORTING container_name = 'CUSTOM'  
           lifetime       = lifetime.
```

4. Erzeugen Sie das Picture Control. Für dieses können Sie auch eine Lebensdauer festlegen. Allerdings darf die Lebensdauer nicht größer sein als die seines Containers.

```
CREATE OBJECT picture  
  EXPORTING parent = container  
           lifetime = lifetime.
```

Ereignis anmelden

5. Das Anmelden von Ereignissen gliedert sich in das Registrieren des Ereignisses am Control Framework, das Definieren einer Behandlungsmethode und das Anmelden der Behandlungsmethode. Diese Schritte finden Sie in [Registrieren und Verarbeiten von Ereignissen \[Seite 12\]](#).

Arbeiten mit dem Control

6. Diese Schritte sind control-spezifisch und werden hier nicht beschrieben.

Abbau der Controls

In der Regel übernimmt das [Lifetime Management \[Extern\]](#) den Abbau der verwendeten Controls. Wenn Sie die Controls aber in Ihrem Programm selbst abbauen wollen, führen Sie folgende Schritte aus:

Anlegen eines Controls am Beispiel des SAP Picture

7. Bauen Sie das Custom Control mit der Methode [free \[Seite 35\]](#) am Frontend ab. Sofern Sie den Control Container nicht mehr benötigen, bauen Sie auch diesen ab:

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



Beachten Sie die Reihenfolge, in der Sie die Controls am Frontend abbauen. Wenn Sie einen Container abbauen, werden nämlich automatisch alle in dem Container eingebundenen Controls auch abgebaut. Der Versuch, ein bereits abgebautes Control erneut abzubauen, führt zu einem Fehler. Mit der Methode [is_alive \[Seite 40\]](#) kann nachgeprüft werden, ob das Control schon abgebaut wurde.

8. Löschen Sie die Referenzvariablen des Custom Controls und des Control Containers:

```
FREE PICTURE.
FREE CONTAINER.
```

Arbeiten mit dem SAP Picture

In diesem Abschnitt werden die für das SAP Picture spezifischen Funktionen aufgeführt.

Voraussetzungen

Der in diesem Abschnitt beschriebene Prozeß stellt nur eine control-spezifische Ergänzung des [allgemeinen Prozesses zur Control-Einbindung \[Seite 8\]](#) dar und ist isoliert nicht lauffähig.

Ablauf



Die Code-Abschnitte sind Beispiele, die nicht immer den vollen Funktionsumfang ausnutzen. Genauere Informationen finden Sie immer im Referenzteil dieser Dokumentation.

Instanziieren

1. Definieren Sie eine Referenzvariable für das SAP Picture:

```
DATA picture TYPE REF TO cl_gui_picture.
```

2. [Erzeugen Sie eine Instanz \[Seite 20\]](#) des SAP Picture:

```
CREATE OBJECT picture
    EXPORTING parent = container.
```

Ereignis anmelden

3. Melden Sie Ereignisse für das SAP Picture an. Folgende Ereignisse unterstützt das SAP Picture:

Ereignisname	Bedeutung
EVENTID_PICTURE_CLICK	Einfachklick auf die Grafik
EVENTID_PICTURE_DBLCLICK	Doppelklick auf die Grafik
EVENTID_CONTROL_CLICK	Einfachklick auf Control-Bereich
EVENTID_CONTROL_DBLCLICK	Doppelklick auf Control-Bereich
EVENTID_CONTEXT_MENU	Klick mit rechter Maustaste (bzw. SHIFT F10) zum Definieren eines Kontextmenüs
EVENTID_CONTEXT_MENU_SELECTED	Auswahl eines Menüpunktes eines Kontextmenüs

Arbeiten mit dem Control

4. Laden Sie eine [Grafik \[Seite 21\]](#) oder eine [SAP-Ikone \[Seite 24\]](#):

```
CALL METHOD picture->load_picture_from_url
    EXPORTING url = 'http://www.xxx.de/images/picture.gif'
    EXCEPTIONS error = 1 .
```

5. [Definieren \[Seite 23\]](#) Sie die Art und Weise, wie die Grafik angezeigt werden soll.

Arbeiten mit dem SAP Picture

```
CALL METHOD picture->set_display_mode
    EXPORTING display_mode = picture->display_mode_stretch
    EXCEPTIONS error = 1.
```

6. [Löschen \[Seite 22\]](#) Sie eventuell die Grafik wieder aus der Anzeige:

```
CALL METHOD picture->clear_picture.
```

Abbau der Controls

In der Regel übernimmt das [Lifetime Management \[Extern\]](#) den Abbau der verwendeten Controls. Wenn Sie die Controls aber in Ihrem Programm selbst abbauen wollen, führen Sie folgende Schritte aus:

7. Bauen Sie das Custom Control am Frontend ab. Sofern Sie den Control Container nicht mehr benötigen, bauen Sie auch diesen ab:

```
CALL METHOD picture->free.
```

8. Löschen Sie die Referenzvariablen des SAP Picture und des Control Containers:

```
FREE PICTURE.
```

Registrieren und Verarbeiten von Ereignissen

Registrieren und Verarbeiten von Ereignissen

Einsatzmöglichkeiten

Über den Ereignismechanismus des Control Frameworks können Sie in Ihrem Programm in Behandlermethoden auf Ereignisse reagieren, die auf dem Control ausgelöst wurden (z.B. Doppelklick).

Voraussetzungen

Der Ablauf wurde allgemeingültig für alle Custom Controls gehalten. Control-spezifische Informationen finden Sie in der jeweiligen Dokumentation zu der Control-Verschaltung.

Ablauf

1. Es wird davon ausgegangen, daß Sie mit einem Custom Control mit der Verschaltung `cl_gui_xyz` arbeiten:

```
DATA my_control TYPE REF TO cl_gui_xyz.
```

Anmelden von Ereignissen beim Control Framework

2. Definieren Sie eine interne Tabelle (Typ `cntl_simple_events`) und einen dazugehörenden Arbeitsbereich (Typ `cntl_simple_event`).

```
DATA events TYPE cntl_simple_events.
DATA wa_events TYPE cntl_simple_event.
```

3. Füllen Sie nun die Ereignistabelle mit den relevanten Ereignissen. Dazu benötigen Sie die Identifikation des Ereignisses (`event_id`). Diese Information finden Sie wiederum in der Klassenbibliothek in den Attributen der Klasse `cl_gui_xyz`. Weiterhin müssen Sie sich entscheiden, ob das Ereignis als Systemereignis (`appl_event = ' '`) oder als Applikationsereignis (`appl_event = 'X'`) definiert werden soll.

```
wa_events-eventid = event_id.
wa_events-appl_event = appl_event.
APPEND wa_events TO events.
```

4. Im nächsten Schritt muß die Ereignistabelle an das Frontend geschickt werden, damit die relevanten Ereignisse vom Frontend an das Backend weitergeleitet werden.

```
CALL METHOD my_control->set_registered_events
         events = events.
```

Um auf ein Ereignis Ihres Custom Controls eingehen zu können, müssen Sie nun noch eine Behandlermethode angeben. Die Behandlermethode kann eine Instanzmethode oder eine statische Methode sein:

Behandlung des Ereignisses als Instanzmethode

5. Definieren Sie die (lokale) Klassendefinition für den Ereignisbehandler. Dabei legen Sie den Namen der Ereignisbehandlermethode fest (`Event_Handler`). Als Information müssen Sie sich in der Klassenbibliothek für die Klasse des Custom Controls `cl_gui_xyz` den Namen des Ereignisses (`event_name`) und die zu diesem Ereignis gehörenden Ereignisparameter (`event_parameter`) besorgen. Der Ereignisparameter `sender` wird bei jedem Ereignis zur

Registrieren und Verarbeiten von Ereignissen

Verfügung gestellt. Der Parameter enthält die Referenz des Controls, das dieses Ereignis ausgelöst hat.

```
CLASS lcl_event_receiver DEFINITION.
PUBLIC SECTION.
METHODS Event_Handler
    FOR EVENT event_name OF cl_gui_xyz
    IMPORTING event_parameter
            sender.
ENDCLASS.
```

- Melden Sie jetzt noch für die registrierten Ereignisse Behandlungsmethoden beim OO Control Framework an.

```
DATA event_receiver TYPE REF TO lcl_event_receiver.
CREATE OBJECT event_receiver.
SET HANDLER event_receiver->Event_Handler
    FOR my_control.
```

Registrieren als statische Methode

- Definieren Sie die (lokale) Klassendefinition für den Ereignisbehandler. Dabei legen Sie den Namen der statischen Ereignisbehandlungsmethode fest (**Event_Handler**). Als Information müssen Sie sich in der Klassenbibliothek für die Klasse des Custom Controls **cl_gui_xyz** den Namen des Ereignisses (**event_name**) und die zu diesem Ereignis gehörenden Ereignisparameter (**event_parameter**) besorgen.

```
CLASS lcl_event_receiver DEFINITION.
PUBLIC SECTION.
CLASS-METHODS Event_Handler
    FOR EVENT event_name OF cl_gui_xyz
    IMPORTING event_parameter
            sender.
ENDCLASS.
```

- Melden Sie jetzt noch für die registrierten Ereignisse Behandlungsmethoden beim OO Control Framework an.

```
SET HANDLER lcl_event_receiver=>Event_Handler
    FOR my_control.
```

Verarbeiten von Control-Ereignissen

- Wie Sie auf ein Ereignis reagieren wollen, legen Sie im Implementierungsteil der Behandlungsmethode fest.

```
CLASS lcl_event_receiver IMPLEMENTATION.
METHOD Event_Handler.
* Abarbeitung des Ereignisses
ENDMETHOD.
ENDCLASS.
```

- Je nach Art des Ereignisses (System- oder Applikationsereignis) müssen Sie jetzt noch die Methode **CL_GUI_CFW=>DISPATCH** aufrufen. Lesen Sie dazu [Ereignisbehandlung \[Extern\]](#).

Ereignisse des SAP Picture

Ereignisse des SAP Picture

Verwendung

Durch bestimmte Benutzerinteraktionen auf dem SAP Picture werden Ereignisse ausgelöst:

Ereignis	Ereignis-ID	Bedeutung
	picture->	
picture_click	eventid_picture_click	Einfachklick auf die Grafik
picture_dblick	eventid_picture_dblick	Doppelklick auf die Grafik
control_click	eventid_control_click	Einfachklick auf Control-Bereich
control_dblick	eventid_control_dblick	Doppelklick auf Control-Bereich
context_menu	eventid_context_menu	Klick mit rechter Maustaste (bzw. SHIFT F10) zum Definieren eines Kontextmenüs
context_menu_selected	eventid_context_menu_selected	Auswahl eines Menüpunktes eines Kontextmenüs

Manche Ereignisse liefern Ereignisparameter mit:

Ereignis	Parameter	Bedeutung
picture_click	MOUSE_POS_X MOUSE_POS_Y	Position der Maus beim Mausklick relativ zum Bild. Die Position wird in bezug auf die Originalgröße des Bildes verwendet.
picture_dblick	MOUSE_POS_X MOUSE_POS_Y	Position der Maus beim Mausklick Mausklick relativ zum Bild. Die Position wird in bezug auf die Originalgröße des Bildes verwendet.
control_click	MOUSE_POS_X MOUSE_POS_Y	Position der Maus beim Mausklick relativ zum SAP Picture Control
control_dblick	MOUSE_POS_X MOUSE_POS_Y	Position der Maus beim Mausklick relativ zum SAP Picture Control
context_menu		
context_menu_selected	FCODE	OK_CODE des ausgewählten Kontextmenüs

Integration

Wenn Sie auf diese Ereignisse in Ihrem ABAP-Programm reagieren müssen, müssen Sie sich auf diese Ereignisse registrieren. Dazu verwenden Sie die Methode [set_registered_events \[Seite 38\]](#). Ereignisse, auf die Sie sich nicht registriert haben, werden schon am Frontend ausgefiltert und gelangen nicht zum Backend.

Siehe auch: [Ereignisbehandlung \[Extern\]](#)

Aktivitäten

Lesen Sie den allgemeinen [Prozeß \[Seite 12\]](#) für das Arbeiten mit Ereignissen des Control Frameworks.

Verwendung von Controls im WAN

Verwendung von Controls im WAN

Die Verwendung von Controls zur Oberflächengestaltung führt in der Regel zu einer Belastung des Kommunikationskanals zwischen Frontend und Backend. Dies kann schon im LAN-, aber insbesondere im WAN-Umfeld ein performance-kritischer Aspekt sein.

Puffermechanismen helfen, diese Problematik zu entschärfen (siehe auch [Automation Queue \[Extern\]](#)). Die aufgeführten Punkte sollen als Richtlinien bei der Verwendung von Controls im WAN dienen.

Control-spezifische Hinweise zur Verwendung im WAN finden Sie in der Dokumentation zu dem jeweiligen Control.

Verwendung von CL_GUI_CFW=>FLUSH

Der Aufruf [CL_GUI_CFW=>FLUSH \[Seite 29\]](#) dient zum Synchronisieren der Automation Queue und der in der Queue enthaltenen ABAP-Variablen. Dieser Aufruf erzeugt in vielen Fällen einen synchronen RFC vom Applikationsserver zum Frontend. Um optimale Performance zu erreichen, sollten die Aufrufe dieser Methode minimiert werden.

Es ist in vielen Fällen zu empfehlen, alle Eigenschaften von allen Controls zentral an einer Stelle (z.B. am Anfang des PAI) in einer Automation Queue zu lesen und dann über einen einmalige Synchronisation zu besorgen. Diese Variante ist auch dann zu bevorzugen, wenn dabei Eigenschaften gelesen werden, die nicht immer für den Ablauf des Ereignisbehandlers bzw. des PAI – PBO Zyklus notwendig sind.

Es ist nicht notwendig, einen „Sicherheits-Flush“ am Ende von PBO zu codieren, damit Methodenaufrufe der Controls garantiert an das Frontend transportiert werden. Diese Funktionalität wird systemseitig garantiert, wenn das nächste Dynpro gesendet wird. Damit ist es auch nicht möglich, eine Automation Queue über mehrere Bildwechsel hinweg aufzubauen.

Es ist nicht garantiert, daß eine Automation Queue durch den Aufruf CL_GUI_CFW=>FLUSH gesendet wird. Die Queue erkennt, ob Returnwerte enthalten sind. Ist dies nicht der Fall, wird das Senden unterdrückt!

Für alle Fälle, in denen auch bei einer Queue ohne Returnwert gewünscht wird, daß die Automation Queue synchron versendet wird, gibt es im Control Framework die Methode [CL_GUI_CFW=>UPDATE_VIEW \[Seite 32\]](#). Diese Methode darf nur dann verwendet werden, wenn es zwingend notwendig ist, ein Update des 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.

Nach dem Lesen von Eigenschaften ist der Inhalt der entsprechenden ABAP-Variablen erst nach dem nächsten FLUSH garantiert. Solange dieser Aufruf nicht erfolgt ist, ist der Inhalt der entsprechenden ABAP-Variablen nicht definiert. In Zukunft wird es Fälle gegeben, in denen dieser FLUSH unnötig sein wird. Diese Fälle werden von der Automation Queue erkannt; der entsprechende FLUSH-Call wird dann ignoriert.

Erzeugen von Controls, Datenversorgung

Das Erzeugen eines Controls und die Datenversorgung ist in den meisten Fällen ein einmaliger Vorgang und im Vergleich zu Dynproelementen sehr teuer. Deshalb sollten Controls nicht unnötig erzeugt bzw. nicht unnötig mit Daten versorgt werden.

Ein typisches Beispiel hierfür sind TabStrips mit mehreren Seiten. Wenn diese Seiten Controls tragen, ist immer abzuwägen, ob man auf lokale Seiten verzichtet und die Controls erst dann

Verwendung von Controls im WAN

erzeugt, wenn der Benutzer die Seite aktiviert. Das gleiche trifft für die Datenversorgung dieser Controls auf TabStrip-Seiten zu.

Muß bei der Datenversorgung eine Unterscheidung zwischen einer WAN- und einer LAN-Anmeldung vorgenommen werden, steht der Funktionsbaustein `SAPGUI_GET_WANFLAG` zur Verfügung. In manchen Fällen kann es notwendig werden, daß eine Anwendung andere Datenmengen oder ganze Fallbacks für die WAN-Anmeldung zur Verfügung stellen sollte. Ein Beispiel, wann die WAN- bzw. LAN- Anmeldung einen Einfluß haben kann, ist die Anzahl von Geschwistern in einem Tree Control, die ohne künstliche Zwischenebenen übertragen werden können.

Im Gegensatz zu Dynproelementen werden die Controls nur einmalig erzeugt und mit Daten versorgt. Controls werden unter Performance-Aspekten dann immer preiswerter, je länger diese leben. In Anwendungen, die ständig neu aufgerufen und damit neu initialisiert werden, kann dies zu einem erheblichen Performance-Nachteil werden; in Anwendungen, die sehr lange auf den gleichen Bildern arbeiten, kann daraus sogar ein Performance-Vorteil entstehen.

Im Einzelfall kann über entsprechende [Performance-Werkzeuge \[Extern\]](#) überprüft werden, welche Nachteile oder Vorteile die Verwendung eines Controls unter dem Aspekt der Netzwerkauslastung bringt.

Ablegen von Dokumenten, Bildern etc.

Zum Release 4.6A wird ein Frontend-Cache für Zugriffe auf Dokumente aus dem BDS (Business Dokument Service) realisiert. Es wird dringend empfohlen, Office-Dokumente, Bilder etc. im BDS und nicht in der R/3-Datenbank abzulegen. Dokumente aus dem BDS können danach im Frontend-Cache abgelegt werden. Sie müssen nur einmalig über das Netz geladen werden.

Spezielle Hinweise für das SAP Picture

Spezielle Hinweise für das SAP Picture

Zusätzlich zu den Hinweisen, die für alle Controls gültig sind, müssen Sie folgendes beachten:

Spezielle Beachtung muß im WAN den Bildern geschenkt werden, die mit dem Data Provider über **SAPR3** URLs geladen werden. Diese URLs werden von dem Programmierer mit dem Funktionsbaustein **DP_CREATE_URL** erzeugt. In diesem Fall sollte darauf geachtet werden, daß die Gesamtgröße aller Bilder, die gleichzeitig angezeigt werden, möglichst 30 Kilobyte nicht übersteigt. Deswegen sind immer die Formate GIF oder JPG gegenüber BMP zu bevorzugen. Die Formate GIF und JPG sind komprimiert und verbrauchen weniger Speicher.

Das gleiche gilt, wenn die Bilder von einem WEB Server geladen werden, der über eine WAN-Leitung angesprochen wird.

Methoden der Klasse CL_GUI_PICTURE

Diese Klasse beinhaltet sowohl controlspezifische Methoden, als auch vom Control Framework geerbte Methoden. In diesem Abschnitt werden nur die controlspezifischen Methoden beschrieben. Die relevanten Methoden des Control Frameworks werden in [Methoden des OO Control Frameworks \[Seite 26\]](#) beschrieben.

constructor

constructor

Diese Methode wird für die Instanzierung eines SAP Picture verwendet.

```
CREATE OBJECT picture
  EXPORTING lifetime = lifetime
           shellstyle = shellstyle
           parent = parent
  EXCEPTIONS ERROR = 1.
```

Parameter	Bedeutung
lifetime	<p>Parameter für das Lifetime Management [Extern]. Folgende Werte sind möglich:</p> <p>picture->lifetime_imode: 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 37] aufgerufen</p> <p>picture->lifetime_dynpro: Das Control lebt, solange die Instanz des Dynpros existiert, d.h. sich noch im Dynprostapel befindet. Danach wird die Methode free [Seite 35] aufgerufen.</p> <p>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>picture->lifetime_default: 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 ost), dann wird die Lebensdauer auf <code>picture->lifetime_imode</code> gesetzt.</p>
shellstyle	<p>Steuerung des Erscheinungsbilds und des Verhaltens des Controls</p> <p>Konstanten aus dem ABAP-Include <CTLDEF>, die mit WS beginnen, können Sie übergeben. Kombinationen von mehreren Styles können Sie durch Addieren der Konstanten erreichen. Der Default-Wert führt intern zum Setzen einer ausreichenden Kombination von Style-Konstanten.</p>
parent	<p>Container, in dem das SAP Picture angezeigt werden kann (Siehe SAP Container [Extern])</p>

load_picture_from_url

Mit dieser Methode bringen Sie eine Grafik im Format gif, wmf oder jpg im SAP Picture zur Anzeige.

```
CALL METHOD picture->load_picture_from_url
    EXPORTING url = url
    IMPORTING result = result
    EXCEPTIONS error = 1.
```

Parameter	Bedeutung
url	Adresse, unter der das Bild zu finden ist. Die Adresse kann eine HTTP-oder FTP-Adresse sein. Weiterhin werden Adressen auf einem File-System (FILE) und vom SAP Data Provider [Extern] generierte Adressen im Speicher des Frontend unterstützt.
result	0: Fehler beim Laden des Bilds 1: Bild wurde erfolgreich geladen

clear_picture

clear_picture

Diese Methode entfernt das aktuelle Bild aus der Anzeige.

```
CALL METHOD picture->clear_picture  
EXCEPTIONS error = 1.
```

set_display_mode

Mit dieser Methode können Sie die Art einstellen, wie die Grafik angezeigt werden soll.

```
CALL METHOD picture->set_display_mode
    EXPORTING display_mode = display_mode
    EXCEPTIONS error = 1.
```

Parameter	Bedeutung
display_mode	<p>Parameter für die Anzeigeart. Folgende Werte sind möglich:</p> <p>picture->display_mode_stretch: Die Grafik wird so vergrößert/verkleinert, daß sie genau in den Bereich des SAP Picture paßt.</p> <p>picture->display_mode_fit: Die Grafik wird so vergrößert, daß sie vertikal und horizontal gleichmäßig vergrößert wird und in mindestens einer Richtung genau in das SAP Picture paßt.</p> <p>picture->display_mode_normal: Die Grafik wird in ihrer Originalgröße angezeigt.</p> <p>picture->display_mode_fit_center: Wie picture->display_mode_fit, allerdings wird die Grafik zusätzlich in der Mitte des Control-Bereichs angeordnet.</p> <p>picture->display_mode_normal_center: Wie picture->display_mode_normal, allerdings wird die Grafik zusätzlich in der Mitte des Control-Bereichs angeordnet.</p>

`load_picture_from_sap_icons`

load_picture_from_sap_icons

Diese Methode bringt eine SAP-Ikone zur Anzeige.

```
CALL METHOD picture->load_picture_from_sap_icons
      EXPORTING icon = icon
      EXCEPTIONS error = 1.
```

Parameter	Bedeutung
icon	Name der SAP-Ikone



Die Ikone kann über ihren Namen (z.B. `ICON_ANNOTATION`) angesprochen werden. Dazu muß aber in Ihrem Programm das Include `<ICON>` eingebunden werden.

Ansonsten sprechen Sie die Ikonen über `@xy@` an, wobei `xy` die entsprechende Buchstabenkombination darstellt.

display_context_menu

Diese Methode verwenden Sie, um innerhalb der Ereignisbehandlungsmethode für das Ereignis `context_menu` das erstellte Menü `menu` zur Anzeige zu bringen.

```
CALL METHOD picture->display_context_menu
      exporting context_menu = context_menu
      exceptions error = 1.
```

Parameter	Bedeutung
<code>context_menu</code>	Instanz der Klasse <code>CL_CTMENU</code> für das Aufbauen und Anzeigen von Kontextmenüs (siehe auch: Kontextmenü [Extern]).

Methoden des OO Control Frameworks

Methoden des OO Control Frameworks

In diesem Abschnitt werden relevante Methoden des OO Control Frameworks beschrieben, die Sie bei der Implementierung des SAP Picture benötigen.

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 \[Seite 29\]](#) 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_OBJECT

Die Klasse `CL_GUI_OBJECT` beinhaltet wichtige Methoden zum Verschalen von Custom Controls. Für Anwendungsprogramme ist einzig die Methode [is_valid \[Seite 34\]](#) relevant.

is_valid

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

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.
```

Methoden der Klasse CL_GUI_CONTROL**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.

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 \[Seite 35\]](#) aufgerufen, bevor das Control am Frontend abgebaut wird.

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

set_registered_events**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 my_control

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.

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 \[Extern\]](#).

is_alive

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

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
<code>my_control->align_at_left</code>	Ausrichtung am linken Rand
<code>my_control->align_at_right</code>	Ausrichtung am rechten Rand
<code>my_control->align_at_top</code>	Ausrichtung am oberen Rand
<code>my_control->align_at_bottom</code>	Ausrichtung am unteren Rand

Kombinationen erhält man durch Aufaddieren der Komponenten:

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

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_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

get_focus**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

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

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_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