

SAP TextEdit



HELP.BCCITEDIT

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 TextEdit	7
Verwendung von Controls im WAN	10
Spezielle Hinweise für das SAP TextEdit	12
Methoden	13
alphabetische Sortierung.....	14
AUTO_REDRAW	15
COMMENT_LINES.....	16
COMMENT_SELECTION.....	17
CONSTRUCTOR.....	18
DELETE_TEXT	21
EMPTY_UNDO_BUFFER	22
FIND_AND_REPLACE.....	23
FIND_AND_SELECT_TEXT	25
GET_FIRST_VISIBLE_LINE	27
GET_LINE_TEXT	28
GET_PATH_OF_DROPPED_FILES.....	29
GET_SELECTED_TEXT_AS_R3TABLE	30
GET_SELECTED_TEXT_AS_STREAM.....	31
GET_SELECTION_INDEXES.....	32
GET_SELECTION_POS	33
GET_TEXTMODIFIED_STATUS	34
GET_TEXT_AS_R3TABLE	35
GET_TEXT_AS_STREAM.....	37
GO_TO_LINE	39
HIGHLIGHT_BREAKPOINT_LINE.....	40
HIGHLIGHT_LINES.....	41
HIGHLIGHT_SELECTION.....	42
INDENT_LINES.....	43
INDENT_SELECTION.....	44
MAKE_SELECTION_VISIBLE	45
OPEN_LOCAL_FILE	46
PROTECT_LINES	47
PROTECT_SELECTION.....	48
REGISTER_EVENT_CONTEXT_MENU	49
REGISTER_EVENT_DBLCLICK	50
REGISTER_EVENT_F1.....	51
REGISTER_EVENT_F4.....	52
REGISTER_EVENT_FILEDROP	53
REPLACE_ALL	54
SAVE_AS_LOCAL_FILE.....	56
SELECT_LINES	57
SET_AUTOINDENT_MODE	58
SET_COMMENTS_STRING.....	59
SET_DRAGDROP	60
SET_FILEDROP_MODE.....	61
SET_FIRST_VISIBLE_LINE.....	62
SET_HIGHLIGHT_BREAKPOINTS_MODE	63
SET_HIGHLIGHT_COMMENTS_MODE	64

SET_LOCAL_CONTEXTMENU_MODE	65
SET_NAVIGATE_ON_DBLCLICK	66
SET_READONLY_MODE	67
SET_SELECTED_TEXT_AS_R3TABLE	68
SET_SELECTED_TEXT_AS_STREAM	69
SET_SELECTION_INDEXES	70
SET_SELECTION_POS	71
SET_SELECTION_POS_IN_LINE	72
SET_SPACES_ON_INDENT	73
SET_STATUSBAR_MODE	74
SET_STATUS_TEXT	75
SET_TEXTMODIFIED_STATUS	76
SET_TEXT_AS_R3TABLE	77
SET_TEXT_AS_STREAM	78
SET_TOOLBAR_MODE	79
SET_WORDBREAK_PROCEDURE	80
SET_WORDWRAP_BEHAVIOR	81
UNCOMMENT_LINES	82
UNCOMMENT_SELECTION	83
UNINDENT_LINES	84
UNINDENT_SELECTION	85
funktionale Sortierung	86
Control anlegen	87
Setzen und Holen von Text	88
Erfragen und Setzen von Textpositionen	89
Hervorheben und Schützen von Textteilen	91
Suchen und Ersetzen von Textteilen	92
Statusbar	93
Toolbar	94
Sonstige	95
Ereignisse	96
Klassenkonstanten	97
Instanz-Attribute	98
Tastatur- und Maussteuerung im Editorfenster	102
Methoden der Klasse CL_GUI_CFW	106
dispatch	107
flush	108
get_living_dynpro_controls	109
set_new_ok_code	110
update_view	111
Methoden der Klasse CL_GUI_OBJECT	112
is_valid	113
free	114
Methoden der Klasse CL_GUI_CONTROL	115
constructor	116
finalize	118
set_registered_events	119
get_registered_events	120
is_alive	121

set_alignment.....	122
set_position.....	123
set_visible.....	124
get_focus.....	125
set_focus.....	126
get_height.....	127
get_width.....	128
Methoden der Klasse CL_DRAGDROP.....	129
constructor.....	130
add.....	131
clear.....	133
destroy.....	134
get.....	135
get_handle.....	137
modify.....	138
remove.....	140
Methoden der Klasse CL_DRAGDROPOBJECT.....	141
set_flavor.....	142
abort.....	143

SAP TextEdit

Verwendung

Mit dem SAP TextEdit realisieren Sie einen Editor, der zur Eingabe und Bearbeitung von reinem Text dient. Als Anwendungsgebiet ist der Einsatz als einfacher Multiline-Editor und als ABAP-Quelltexteditor vorgesehen. Mit Hilfe von temporären Zusätzen können Sie zusätzliche Darstellungsmöglichkeiten, z.B. das Hervorheben von Texten, verwenden. Temporär bedeutet in diesem Zusammenhang, daß die Zusätze lediglich auf dem Frontend gesetzt und verwaltet, aber nicht an den jeweiligen Controller zurückgegeben werden. Das Zuweisen von Formatinformationen oder das Einbinden von Grafiken ist nicht vorgesehen.

Funktionsumfang

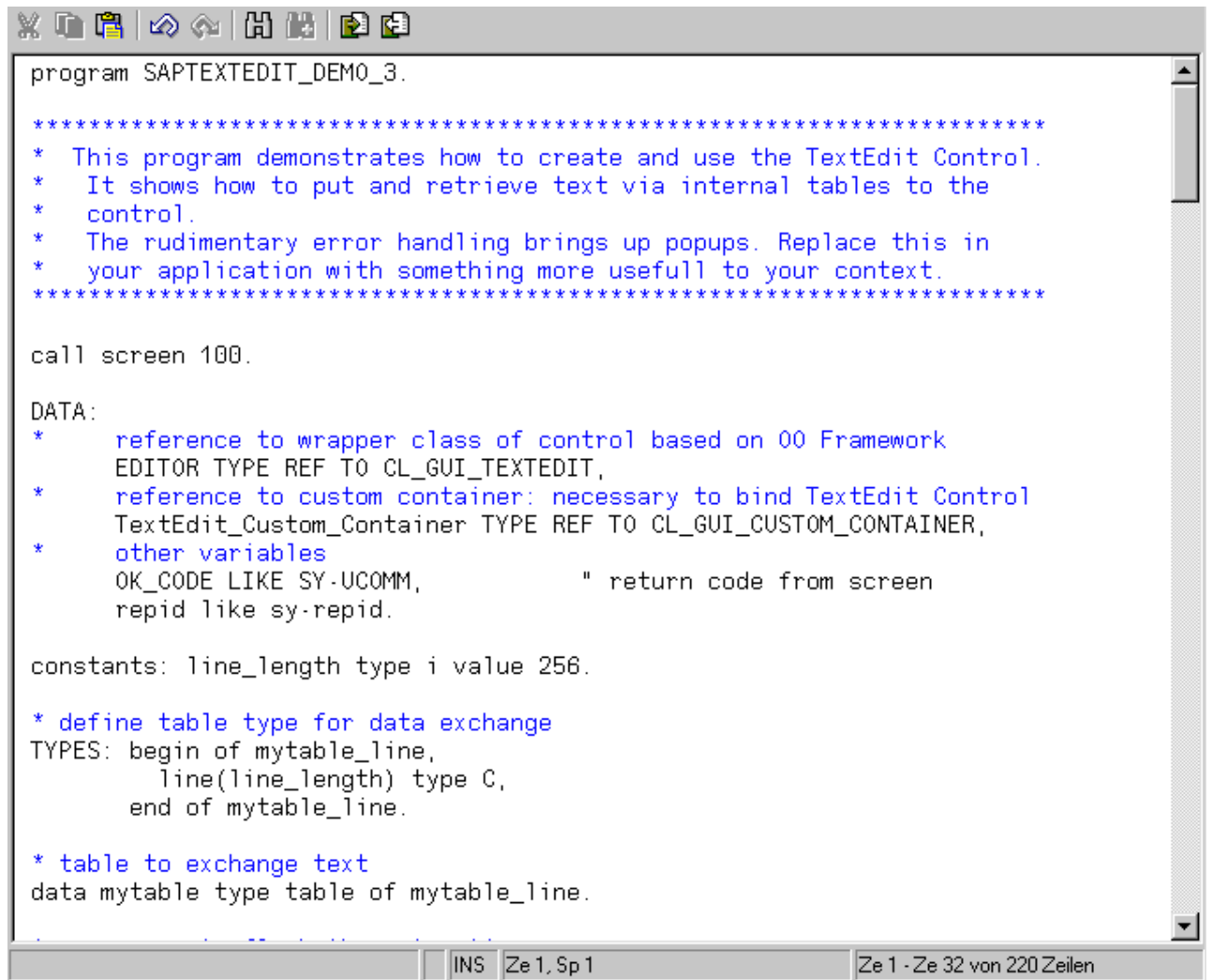
Das SAP TextEdit besteht aus drei Einheiten:

- der Drucktastenleiste mit vordefinierten Ikonen
- dem Editorfenster zum Darstellen des Textes
- der Statusleiste bestehend aus den fünf Feldern
 - Ausgabe von Textmeldungen
 - Angaben zu den selektierten Textbereichen
 - aktuelle Cursorposition und Gesamtanzahl der Zeilen
 - Änderungsstatus (*' geändert, ' ' unverändert)
 - Einfüge- oder Überschreibmodus ('Ins' oder 'Ovr')

Die Darstellung der Drucktastenleiste und Statusleiste ist optional.

Screenshot eines SAP TextEdit

SAP TextEdit



```

program SAPTEXTEDIT_DEMO_3.

*****
* This program demonstrates how to create and use the TextEdit Control.
* It shows how to put and retrieve text via internal tables to the
* control.
* The rudimentary error handling brings up popups. Replace this in
* your application with something more usefull to your context.
*****

call screen 100.

DATA:
* reference to wrapper class of control based on OO Framework
EDITOR TYPE REF TO CL_GUI_TEXTEDIT,
* reference to custom container: necessary to bind TextEdit Control
TextEdit_Custom_Container TYPE REF TO CL_GUI_CUSTOM_CONTAINER,
* other variables
OK_CODE LIKE SY-UCOMM,           " return code from screen
repid like sy-repid.

constants: line_length type i value 256.

* define table type for data exchange
TYPES: begin of mytable_line,
        line(line_length) type C,
        end of mytable_line.

* table to exchange text
data mytable type table of mytable_line.

```

INS Ze 1, Sp 1 Ze 1 - Ze 32 von 220 Zeilen

Die implementierten Funktionen sind:

- Übergabe bzw. Holen des Textes in Form von Tabellen an das bzw. von dem Control
- Anzeige- (Read-Only-) und Änderungsmodus
- Einfüge- und Überschreibmodus
- Einstellung der maximalen Zeilenlänge
- Hervorheben von Textbereichen (auch mit einem Prefix am Zeilenbeginn) sowie deren Sperrung gegen Eingabe
- Abfrage der aktuellen Cursorposition und der Position eines markierten Bereiches
- Abfrage und Setzen der ersten im Editor angezeigten Zeile
- unterschiedliches Verhalten des Zeilenumbruches

- Code-Editor:
Der Umbruch erfolgt zeilenorientiert mit einstellbarer maximalen Zeilenlänge. Die technische Realisierung erfolgt über die Abbildung von internen Tabellen.
- Fließtexteditor:
 - a) mit Orientierung der Zeilenlänge an der Fenstergröße
 - b) ohne Begrenzung der Zeilenlänge und ohne Umbrüche
- Lokales Kontextmenü mit den Funktionen der Toolbar
- Optionale Toolbar für die im Control vorhandenen lokalen Operationen
- einfache Navigation im Text durch eine Vielzahl von Tastaturkommandos. Die optionale Drucktastenleiste vereinfacht lokale Operationen
- Cut, Copy&Paste für markierte Textbereiche, erfolgt entweder über die Tastatur oder die Drucktastenleiste
- Einrücken von markierten Textblöcken
- Laden bzw. Sichern von lokalen Dateien (Import/Export)
- Mehrstufige Undo bzw. Redo
- Suchen und Ersetzen mit den Optionen:
 - (Teil-)String
 - ganzes Wort
 - mit oder ohne Berücksichtigung der Groß- und Kleinschreibung
 - in einem Zeilenbereich
 - Weitersuchen nach einem Treffer
- Feuern des Doppelklick-, F1-, F4-, Drag&Drop- und Kontextmenu-Events
- unterschiedliches Verhalten beim Drag&Drop von Dateien:
 - Anzeigen der Datei ohne Auslösung eines Ereignisses
 - nur eine Datei kann mit Drag&Drop geholt werden und ein Ereignis wird ausgelöst
 - mehrere Dateien können mit Drag&Drop geholt werden und das zugehörige Ereignis wird ausgelöst
- Status, ob ein Text geändert wurde oder nicht

Die ABAP-OO-Verschaltung ist mit der globalen Klasse CL_GUI_TEXTEDIT in der Entwicklungsklasse SAPTEXTEDIT realisiert.



In der Entwicklungsklasse SAPTEXTEDIT befinden sich Beispiel- und Testprogramme.

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 \[Seite 108\]](#) 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 \[Seite 111\]](#). 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 TextEdit

Spezielle Hinweise für das SAP TextEdit

Modifizierter Text wird beim SAP TextEdit immer als ganzer Text synchron ans bzw. vom Frontend übertragen. Bei großen Datenmengen kann damit im WAN, abhängig von der zur Verfügung stehenden Übertragungsbandbreite, der Zeitaufwand für die Übertragung hoch werden.

Generell sind solche Methoden betroffen, die Tabellen als Parameter übertragen. Dies sind folgende:

[SET TEXT AS STREAM \[Seite 78\]](#)

[SET_SELECTED_TEXT_AS_STREAM \[Seite 69\]](#)

[GET TEXT AS STREAM \[Seite 37\]](#)

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

[SET TEXT AS R3TABLE \[Seite 77\]](#)

[SET_SELECTED_TEXT_AS_R3TABLE \[Seite 68\]](#)

[GET TEXT AS R3TABLE \[Seite 35\]](#)

[GET_SELECTED_TEXT_AS_R3TABLE \[Seite 30\]](#)

Das TextEdit verwendet einen Attribute-Cache, in dem gespeichert wird, ob der Text modifiziert wurde, und welches die aktuellen Zeilen - und Spaltenpositionen sind. Das Lesen dieser Attribute ist damit nicht zwingend mit Roundtrips zum Frontend verbunden, sondern erfolgt in vielen Fällen aus dem Attribute-Cache. Der Aufruf [CL_GUI_CFW=>FLUSH \[Seite 108\]](#) ist bei solchen Attribut weiterhin notwendig, da es für den Verwender nicht erkennbar ist, wann die Abfrage aus dem Cache bedient werden kann.

Methoden

[alphabetische Sortierung \[Seite 14\]](#)

[funktionale Sortierung \[Seite 86\]](#)

alphabetische Sortierung**alphabetische Sortierung**

Sie haben die Hilfe zur Anwendung aufgerufen. Zu dem aktuellen R/3-Kontext sind folgende Informationen verfügbar:

AUTO_REDRAW

Verwendung

Diese Methode steuert über die Verwaltung des Referenzzählers M_AUTOREDRAW_REFCOUNTER, ein Read-Only [Instanz-Attribute \[Seite 98\]](#), das erneute Zeichnen des Controls. Erst wenn der Referenzzähler den Wert 0 erreicht hat, zeichnet sich das Control neu. Dies erlaubt das Schachteln von Aufrufen.

Funktionsumfang

call method textedit->auto_redraw

```
exporting  
  enable_redraw = enable_redraw
```

```
exceptions  
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
enable_redraw	Referenzzähler	false Referenzzähler wird hochgezählt true Referenzzähler wird verringert



Diese Methode sollte nur in absolut notwendigen Fällen verwendet werden. Falls ein anderes Fenster über das Textfenster gelegt wird und anschließend der Fokus wieder auf das Textfenster gesetzt wird, so enthält das Textfenster Teile des anderen Fensters. Erst wenn der Referenzzähler den Wert null erreicht hat, wird der Text wieder richtig angezeigt.

COMMENT_LINES

COMMENT_LINES**Verwendung**

Mit dieser Methode können Sie eine Anzahl von Zeilen in Kommentarzeilen umwandeln.

Funktionsumfang

call method textedit->comment_lines

```
exporting
  from_line = from_line
  to_line = to_line
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
from_line	Beginn der Zeilen	
to_line	Ende der Zeilen	
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

COMMENT_SELECTION

Verwendung

Mit dieser Methode können Sie markierte Zeilen in Kommentarzeilen umwandeln. Falls die Start- und Endzeile nur teilweise selektiert wurden, werden die kompletten Zeilen in Kommentarzeilen umgewandelt..

Funktionsumfang

call method textedit->comment_selection

```
exporting
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

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 textedit

exporting

parent = parent

lifetime = lifetime

max_number_chars = max_number_chars

style = style

filedrop_mode = filedrop_mode

wordwrap_mode = wordwrap_mode

wordwrap_position = wordwrap_position

wordwrap_to_linebreak_mode = wordwrap_to_linebreak_mode

exceptions

error_cntl_create = 1

error_cntl_init = 2


error_cntl_link = 3

error_dp_create = 4

gui_type_not_supported = 5.

Parameter	Bedeutung	mögliche Werte
parent	Parent der Instanz; d.h., Container in dem das Control angezeigt wird	
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>

CONSTRUCTOR

max_number_chars	Begrenzt die Anzahl der Zeichen, die maximal in das Control eingegeben werden können	
style	Steuerung des Erscheinungsbild und des Verhaltens des Controls 	Konstanten aus der Klasse CL_GUI_CONTROL (beginnend mit WS_*). Kombinationen von mehrere Styles können Sie durch addieren der Konstanten vollziehen. Der Default-Wert führt intern zum Setzen einer ausreichenden Kombination von Style-Konstanten.
filedrop_mode	Parameter, der das Verhalten beim Drag&Drop steuert	<p>dropfile_event_off Nur eine Datei, die den kompletten existierenden Text ersetzt, kann benutzt werden (Defaultwert).</p> <p>dropfile_event_single Nur eine Datei kann benutzt werden. Die Datei wird nicht in den Editor eingebunden, sondern das Datei-Drop-Ereignis wird ausgelöst. Der Pfad kann anhand der Methode GET_PATH_OF_DROPPED_FILES [Seite 29] angegeben werden.</p> <p>dropfile_event_multiple Mehrere Dateien können benutzt werden. Die Dateien werden nicht in den Editor eingebunden, sondern das Datei-Drop-Ereignis wird ausgelöst. Eine Liste der Pfade kann anhand der Methode GET_PATH_OF_DROPPED_FILES [Seite 29] angegeben werden.</p>

CONSTRUCTOR

wordwrap_mode	Verhalten des Zeilenumbruchs	wordwrap_off kein Zeilenumbruch wordwrap_at_windowborder Zeilenumbruch an der Fenstergrenze (Defaultwert) wordwrap_at_fixed_position Zeilenumbruch an einer festen Position Weitere Informationen finden Sie unter Klassenkonstanten [Seite 97] .
wordwrap_position	Falls wordwrap_mode = textedit->wordwrap_at_fixed_position , Position für den automatischen Zeilenumbruch in einer Zeile	Defaultwert: -1
wordwrap_to_linebreak_mode	Umwandlung der weichen in harte Zeilenumbrüche beim Speichern in das R/3-System	false weiche Zeilenumbrüche werden beim Speichern ignoriert true Umwandlung von weiche in harte Zeilenumbrüche beim Speichern



Container haben den Vorteil, daß Sie bereits beim Erzeugen dieser die wesentlichen Eigenschaften zur Darstellung des Controls (Anordnung auf dem Dynpro, Verhalten bei Größenänderung des Fensters, usw.) angeben können. Nutzen Sie diese Möglichkeit nicht, müssen Sie die Eigenschaften über aufwendige Aufrufe der Methode [SET_WINDOW_PROPERTY \[Extern\]](#) setzen.

DELETE_TEXT

Verwendung

Diese Methode löscht den gesamten Text.

Funktionsumfang

call method textedit->delete_text

exceptions

error_cntl_call_method = 1.

EMPTY_UNDO_BUFFER

EMPTY_UNDO_BUFFER

Verwendung

Diese Methode leert den Undo-Buffer des Controls. Danach kann keine vorherige Aktion rückgängig gemacht werden, auch der Aufruf selbst nicht. Der Undo-Buffer wird erst wieder durch nachfolgende Aktionen im Control aufgebaut. Implizit wird der Undo-Buffer beim Schützen von Textbereichen gegen Eingabe, Methode [PROTECT_LINES \[Seite 47\]](#) und [PROTECT_SELECTION \[Seite 48\]](#), gelöscht.

Funktionsumfang

call method textedit->empty_undo_buffer

exceptions

error_cntl_call_method = 1.

FIND_AND_REPLACE

Verwendung

Diese Methode ermöglicht das Suchen und Ersetzen von Text.

Funktionsumfang

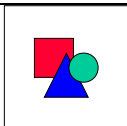
call method textedit->find_and_replace

```
exporting
  case_sensitive_mode = case_sensitive_mode
  replace_string = replace_string
  search_string = search_string
  whole_word_mode = whole_word_mode
```

```
changing
  string_found = string_found
```

```
exceptions
  invalid_parameter = 1
  error_cntl_call_method = 2.
```

Parameter	Bedeutung	Mögliche Werte
case_sensitive_mode	Groß- und Kleinschreibung	true beachten false nicht beachten (Defaultwert)
replace_string	Text, der den gefundenen Begriff ersetzt	
search_string	Zu suchender und/oder ersetzender Text	
whole_word_mode	Nur ganze Wörter sollen gefunden werden	true beachten false nicht beachten (Defaultwert)
string_found	Rückgabewert, der angibt, wie oft der Text gefunden wurde	



➡ Falls Sie weder den Parameter REPLACE_STRING noch SEARCH_STRING setzen, wird das Dialogfenster *Suchen-Ersetzen* aufgerufen.

Falls kein Text selektiert wurde, startet die Suche an der aktuellen Position des Cursors. Falls Text selektiert wurde, startet die Suche am zweiten Zeichen des selektierten Bereiches. Die Suche ist dabei nicht auf den selektierten Bereich beschränkt.

FIND_AND_REPLACE

Die Methode springt am Ende des Textes für die weitere Suche an den Textanfang, d.h. der Text kann mehrmals durchlaufen werden.



Falls alle Vorkommnisse eines Such-Strings komplett in einem Text ersetzt werden soll, benutzen Sie die Methode [REPLACE_ALL \[Seite 54\]](#).

FIND_AND_SELECT_TEXT

Verwendung

Diese Methode ermöglicht das Suchen und Selektieren von Text.

Funktionsumfang

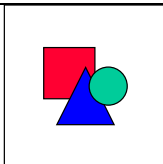
call method textedit->find_and_select_text

exporting
 case_sensitive_mode = case_sensitive_mode
 search_string = search_string
 whole_word_mode = whole_word_mode

changing
 string_found = string_found

exceptions
 invalid_parameter = 1
 error_cntl_call_method = 2.

Parameter	Bedeutung	Mögliche Werte
case_sensitive_mode	Groß- und Kleinschreibung	true beachten false nicht beachten (Defaultwert)
search_string	Zu suchender und/oder ersetzender Text	
whole_word_mode	Nur ganze Wörter sollen gefunden werden	true beachten false nicht beachten (Defaultwert)
string_found	Rückgabewert, der angibt, wie oft der Text gefunden wurde	



➡ Falls Sie den Parameter SEARCH_STRING nicht setzen, wird das Dialogfenster *Suchen* aufgerufen.

Falls kein Text selektiert wurde, startet die Suche an der aktuellen Position des Cursors. Falls Text selektiert wurde, startet die Suche am zweiten Zeichen des selektierten Bereiches. Die Suche ist dabei nicht auf den selektierten Bereich beschränkt.

FIND_AND_SELECT_TEXT

Die Methode springt am Ende des Textes für die weitere Suche an den Textanfang, d.h. der Text kann mehrmals durchlaufen werden.

GET_FIRST_VISIBLE_LINE

Verwendung

Diese Methode holt die Nummer der ersten sichtbaren Zeile im Control.

Funktionsumfang

call method textedit->get_first_visible_line

```
importing  
  line = line
```

```
exceptions  
  error_cntl_call_method = 1.
```

Parameter	Bedeutung
line	Nummer der Zeile. Die Zeilennummerierung beginnt bei eins.

GET_LINE_TEXT

GET_LINE_TEXT

Verwendung

Diese Methode holt den Text in der Zeile mit der Nummer LINE_NUMBER.

Funktionsumfang

call method textedi->get_line_text

exporting

line_number = line_number

importing

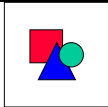
text = text

exceptions

invalid_parameter = 1

error_cntl_call_method = 2.

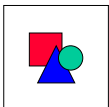
Parameter	Bedeutung
line_number	Nummer der Zeile. Die Zeilennummerierung beginnt bei eins
text	Text der Zeile



Aufgrund zur Zeit noch bestehender Restriktionen bei der Kommunikation zwischen Frontend und Backend, werden Texte lediglich mit einer maximalen Länge von 256 Zeichen übertragen.



Abhilfe für Texte mit mehr als 256 Zeichen:



Selektieren Sie eine ganze Zeile und holen Sie diese dann mit der Methode [GET_SELECTED_TEXT_AS_R3TABLE \[Seite 30\]](#) oder [GET_SELECTED_TEXT_AS_STREAM \[Seite 31\]](#) ab.

GET_PATH_OF_DROPPED_FILES

Verwendung

Diese Methode liefert für beim Drag&Drop benutzte Dateien eine Liste der Pfade. Die Liste wird bei jedem Drag&Drop aktualisiert. Unterschiedliche Drag&Drop Eigenschaften setzen Sie mit der Methode [SET_FILEDROP_MODE \[Seite 61\]](#).

Funktionsumfang

call method textedit->get_path_of_dropped_files

exporting
table = table

exceptions
error_dp = 1.

Parameter	Bedeutung	Mögliche Werte
table	R/3-Tabelle mit den Pfaden	

GET_SELECTED_TEXT_AS_R3TABLE

GET_SELECTED_TEXT_AS_R3TABLE

Verwendung

Diese Methode holt den selektierten Text als R/3-Tabelle.(ohne Informationen zum Zeilenumbruch). Bitte beachten Sie das im Konstruktor ([CONSTRUCTOR \[Seite 18\]](#)) definierte Verhalten des Zeilenumbruchs, insbesondere beim weichen Zeilenumbruch.

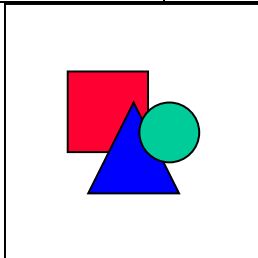
Funktionsumfang

call method textedit->get_selected_text_as_r3table

```
importing  
  table = table
```

```
exceptions  
  error_dp = 1  
  potential_data_loss = 2.
```

Parameter	Bedeutung
table	R/3-Tabelle



Falls Sie eine Tabelle verwenden, deren Zeilenlänge kleiner als die im Control dargestellte ist, so wird der Text lediglich bis zu dieser übergeben. Der restliche Text wird abgeschnitten und geht damit verloren.

Ist die Zeilenlänge der Tabelle größer, so werden Leerzeichen hinzugefügt.

GET_SELECTED_TEXT_AS_STREAM

Verwendung

Diese Methode holt den selektierten Text als R/3-Tabelle in Form eines Streams (mit Informationen zum Zeilenumbruch).

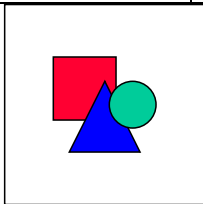
Funktionsumfang

call method textedit->get_selected_text_as_stream

```
importing  
  selected_text = selected_text
```

```
exceptions  
  error_dp = 1.
```

Parameter	Bedeutung
selected_text	R/3-Tabelle



Ein Beispiel zum Umgang mit den Informationen zum Zeilenumbruch finden Sie im Programm SAPTEXTEDIT_TEST_1, Include SAPTEXTEDIT_TEST_1F01, Unterprogramm FIND_CR (Ersetzen des Carrige Return Linefeeds).

GET_SELECTION_INDEXES

GET_SELECTION_INDEXES

Verwendung

Diese Methode liefert Ihnen die Positionsinformationen eines selektierten Textes auf Zeichenebene.

Funktionsumfang

call method textedit->get_selection_indexes

```
importing  
  from_index = from_index  
  to_index = to_index
```

exceptions

```
  error_cntl_call_method = 1.
```

Parameter	Bedeutung
from_index	Zeichen, an der die Selektion beginnt. Der Index ist auf das erste Zeichen im Editor genormt.
to_index	Zeichen, an der die Selektion endet

GET_SELECTION_POS

Verwendung

Diese Methode liefert Ihnen die Positionsinformationen eines selektierten Textes.

Funktionsumfang

call method textedit->get_selection_pos

```
importing
  from_line = from_line
  from_pos = from_pos
  to_line = to_line
  to_pos = to_pos
```

```
exceptions
  error_cntl_call_method = 1.
```


Parameter	Bedeutung
from_line	Zeile, in der die Selektion beginnt
from_pos	Position in der Zeile, an der die Selektion beginnt
to_line	Zeile, in der die Selektion endet
to_pos	Position in der Zeile, an der die Selektion endet

GET_TEXTMODIFIED_STATUS

GET_TEXTMODIFIED_STATUS

Verwendung

Diese Methode liefert Ihnen den Änderungsstatus des Textes. Mit der Methode [SET_TEXTMODIFIED_STATUS \[Seite 76\]](#) können Sie den Änderungsstatus eines Textes setzen.

 Falls Sie Text mit den Methoden [GET_TEXT_AS_R3TABLE \[Seite 35\]](#) und [GET_TEXT_AS_STREAM \[Seite 37\]](#) holen, wird der Änderungsstatus nicht zurückgesetzt.

Funktionsumfang

call method textedit->get_textmodified_status

importing
status = status

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung	Mögliche Werte
status	Änderungsstatus	true Text wurde geändert false Text wurde nicht geändert

GET_TEXT_AS_R3TABLE

Verwendung

Diese Methode holt den gesamten Text im Editor als R/3-Tabelle .(ohne Informationen zum Zeilenumbruch). Bitte beachten Sie das im Konstruktor ([CONSTRUCTOR \[Seite 18\]](#)) definierte Verhalten des Zeilenumbruchs, insbesondere beim weichen Zeilenumbruch.

Funktionsumfang

call method textedit->get_text_as_r3table

exporting
 only_when_modified = only_when_modified

importing
 table = table
 is_modified = is_modified

exceptions
 error_db = 1
 error_cntl_call_method = 2
 error_db_create = 3
 potential_data_loss = 4.

Parameter	Bedeutung	Mögliche Werte
table	R/3-Tabelle	
only_when_modified	Eigenschaft beim Holen des Textes	<p>true Der Text wird nur in die R/3-Tabelle geschrieben, falls der Änderungsstatus des Textes IS_MODIFIED = TRUE ist. Sonst wird eine leere Tabelle übergeben.</p> <p>false Der Text wird grundsätzlich unabhängig vom Änderungsstatus geholt (Defaultwert)</p>

GET_TEXT_AS_R3TABLE

is_modified	Änderungsstatus des Textes	true Text geändert false Text nicht geändert
-------------	----------------------------	--



Falls Sie eine Tabelle verwenden, deren Zeilenlänge kleiner als die im Control dargestellte ist, so wird der Text lediglich bis zu dieser übergeben. Der restliche Text wird abgeschnitten und geht damit verloren.

Ist die Zeilenlänge der Tabelle größer, so werden Leerzeichen hinzugefügt.



Das Setzen des Parameters ONLY_WHEN_MODIFIED auf TRUE erhöht gegebenenfalls die Performance, da der Datenaustausch zwischen Frontend und Backend (und damit zur Datenbank) minimiert wird.

GET_TEXT_AS_STREAM

Verwendung

Diese Methode holt den gesamten Text des Editors als R/3-Tabelle in Form eines Streams (mit Informationen zum Zeilenumbruch).

Funktionsumfang

call method textedit->get_text_as_r3table

exporting
 only_when_modified = only_when_modified

importing
 text = text
 is_modified = is_modified

exceptions
 error_db = 1
 error_cntl_call_method = 2.

Parameter	Bedeutung	Mögliche Werte
text	R/3-Tabelle mit Text	
only_when_modified	Eigenschaft beim Holen des Textes	<p>true Der Text wird nur in die R/3-Tabelle geschrieben, falls der Änderungsstatus des Textes IS_MODIFIED = TRUE ist. Sonst wird eine leere Tabelle übergeben.</p> <p>false Der Text wird grundsätzlich (unabhängig vom Änderungsstatus) geholt (Defaultwert)</p>
is_modified	Änderungsstatus des Textes	<p>true Text geändert</p> <p>false Text nicht geändert</p>



GET_TEXT_AS_STREAM

Das Setzen des Parameters ONLY_WHEN_MODIFIED auf TRUE erhöht gegebenenfalls die Performance, da der Datenaustausch zwischen Frontend und Backend (und damit zur Datenbank) minimiert wird.



Ein Beispiel zum Umgang mit den Informationen zum Zeilenumbruch finden Sie im Programm SAPTEXTEDIT_TEST_1, Include SAPTEXTEDIT_TEST_1F01, Unterprogramm FIND_CR (Ersetzen des carriage return line feeds).

GO_TO_LINE

Verwendung

Diese Methode ermöglicht es, den Cursor in die angegebene Zeile zu navigieren. Die Cursor-Position innerhalb der Zeile ist durch diejenige vor dem Aufruf bestimmt.

Funktionsumfang

call method textedit->go_to_line

exporting
line = line

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung
line	Zeile, zu der navigiert werden soll

HIGHLIGHT_BREAKPOINT_LINE

HIGHLIGHT_BREAKPOINT_LINE

Verwendung

Mit dieser Methode können Sie einen Breakpoint setzen.



Diese Methode wird ausschließlich im ABAP Editor benötigt.

Funktionsumfang

call method textedit->highlight_breakpoint_line

exporting

line = line

highlight_mode = highlight_mode

exceptions

has_no_effect = 1

error_cntl_call_method = 2

invalid_parameter = 3.

Parameter	Bedeutung	Mögliche Werte
line	Zeile, in der der Breakpoint gesetzt werden soll	
highlight_mode	an- oder ausschalten der Hervorhebung (True/False)	true angeschaltet (Defaultwert) false ausgeschaltet

HIGHLIGHT_LINES

Verwendung

Mit dieser Methode können Sie den Modus für die Hervorhebung eines Zeilenbereiches an- oder ausschalten.

Funktionsumfang

call method textedit->highlight_lines

```
exporting
  from_line = from_line
  to_line = to_line
  highlight_mode = highlight_mode
```

```
exceptions
  has_no_effect = 1
  error_cntl_call_method = 2
  invalid_parameter = 3.
```

Parameter	Bedeutung	Mögliche Werte
from_line	Startzeile des Zeilenbereiches	
to_line	Endzeile (inklusive) des Zeilenbereiches	
highlight_mode	an- oder ausschalten der Hervorhebung (True/False)	true angeschaltet (Defaultwert) false ausgeschaltet



Falls Sie das automatische Hervorheben von Kommentarzeilen durch die Methode [SET_HIGHLIGHT_COMMENTS_MODE \[Seite 64\]](#) realisiert haben, bleibt die Methode HIGHLIGHT_LINES wirkungslos.

HIGHLIGHT_SELECTION

HIGHLIGHT_SELECTION

Verwendung

Mit dieser Methode können Sie die Hervorhebung eines markierten Zeilenbereiches an- oder ausschalten.

Funktionsumfang

call method textedit->highlight_selection

exporting
highlight_mode = highlight_mode

exceptions
has_no_effect = 1
error_cntl_call_method = 2
invalid_parameter = 3.

Parameter	Bedeutung	Mögliche Werte
highlight_mode	an- oder ausschalten der Hervorhebung (True/False)	true angeschaltet (Defaultwert) false ausgeschaltet



Falls Sie das automatische Hervorheben von Kommentarzeilen durch die Methode [SET_HIGHLIGHT_COMMENTS_MODE \[Seite 64\]](#) realisiert haben, bleibt die Methode HIGHLIGHT_SELECTION wirkungslos.

INDENT_LINES

Verwendung

Diese Methode rückt einen Zeilenbereich um eine Anzahl von Leerzeichen ein. Die Anzahl setzen Sie mit der Methode [SET_SPACES_ON_INDENT \[Seite 73\]](#) und den Parameter M_SAPCES_ON_INDENT.

Funktionsumfang

call method textedit->indent_lines

```
exporting
  from_line = from_line
  to_line = to_line
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1.
```


Parameter	Bedeutung	Mögliche Werte
from_line	Beginn des Zeilenbereiches, der eingerückt werden soll	
to_line	Ende des Zeilenbereiches, der eingerückt werden soll	
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

INDENT_SELECTION

INDENT_SELECTION

Verwendung

Diese Methode rückt einen markierten Textbereich um eine Anzahl von Leerzeichen ein. Die Anzahl setzen Sie mit der Methode [SET_SPACES_ON_INDENT \[Seite 73\]](#) und den Parameter M_SAPCES_ON_INDENT.

 Es wird nur der tatsächlich markierte Text eingerückt (z.B. falls nicht von Zeilenbeginn an oder bis Zeilenende markiert wurde).

Funktionsumfang

call method textedit->indent_selection

exporting

enable_editing_protected_text = enable_editing_protected_text

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung	Mögliche Werte
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	<p>false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert)</p> <p>true geschützter Textbereich kann bearbeitet werden</p>

MAKE_SELECTION_VISIBLE

Verwendung

Mit dieser Methode können Sie einen selektierten Bereich in den sichtbaren Bereich des Editorfensters verschieben.

Funktionsumfang

call method textedit->make_selection_visible

exceptions

error_cntl_call_method = 1.

OPEN_LOCAL_FILE

OPEN_LOCAL_FILE

Verwendung

Mit dieser Methode können Sie eine lokale Datei öffnen.

Funktionsumfang

call method textedit->open_local_file

exporting
file_name = file_name

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung
file_name	Name der zu öffnenden Datei. Eine Pfadangabe im UNC-Format ist möglich.



Falls Sie den Parameter FILE_NAME nicht setzen, erscheint ein Dialogfenster zur Eingabe des Namens und Pfades der Datei.

PROTECT_LINES

Verwendung

Mit dieser Methode können Sie den Schutz eines Zeilenbereiches gegen Eingabe an- oder ausschalten.

Falls eine Zeile gegen Eingabe geschützt ist, wird die Hintergrundfarbe sichtbar verändert. Bei einer versuchten Eingabe ertönt ein Warnton.

Funktionsumfang

call method textedit->protect_lines

```
exporting
  from_line = from_line
  to_line = to_line
  protect_mode = protect_mode
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1
  invalid_parameter = 2.
```

Parameter	Bedeutung	Mögliche Werte
from_line	Startzeile des Zeilenbereiches	
to_line	Endzeile (inklusive) des Zeilenbereiches	
protect_mode	Schutz gegen Eingabe an- oder ausschalten	true Schutz gegen Eingabe angeschaltet (Defaultwert) false Schutz gegen Eingabe ausgeschaltet
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Text kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

PROTECT_SELECTION

PROTECT_SELECTION

Verwendung

Mit dieser Methode können Sie den Schutz eines markierten Textbereiches gegen Eingabe an- oder ausschalten.

Falls ein Textbereich gegen Eingabe geschützt ist, wird die Hintergrundfarbe sichtbar verändert. Bei einer versuchten Eingabe ertönt ein Warnton.

Funktionsumfang

call method textedit->protect_selection

```
exporting
  protect_mode = protect_mode
```

```
exceptions
  error_cntl_call_method = 1
  invalid_parameter = 2.
```

Parameter	Bedeutung	Mögliche Werte
protect_mode	Schutz gegen Eingabe an- oder ausschalten	true Schutz gegen Eingabe angeschaltet (Defaultwert) false Schutz gegen Eingabe ausgeschaltet
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Text kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

REGISTER_EVENT_CONTEXT_MENU

Verwendung

Mit dieser Methode wird das Event Kontext Menü registriert.



Für ein transparentes Coding wird empfohlen, diese Methode direkt nach der Erzeugung der Instanz (`create object`) aufzurufen.

Funktionsumfang

call method textedit->register_event_context_menu

exporting
 register = register
 appl_event = appl_event
 local_entries = local_entries

exceptions
 error_regist_event = 1
 error_unregist_event = 2
 cntl_error = 3
 event_already_registered = 4
 event_not_registered = 5.

Parameter	Bedeutung	Mögliche Werte
register	Registrierung des Ereignisses	true Ereignis registrieren (Defaultwert) false Ereignis nicht registrieren
appl_event	Auswahl zwischen System- und Anwendungsereignis	space Systemereignis (Defaultwert) 'x' Anwendungsereignis
local_entries	Anzeige der lokalen Einträge im Kontextmenü	true Einträge anzeigen (Defaultwert) false Einträge nicht anzeigen

REGISTER_EVENT_DBLCLICK

REGISTER_EVENT_DBLCLICK

Verwendung

Mit dieser Methode wird das Event [DBLCLICK \[Extern\]](#) (Doppelklick) registriert.



Für ein transparentes Coding wird empfohlen, diese Methode direkt nach der Erzeugung der Instanz (`create object`) aufzurufen.

Funktionsumfang

call method textedit->register_event_dblick

exporting

register = register

appl_event = appl_event

navigate_on_dblick = navigate_on_dblick

exceptions

error_regist_event = 1

error_unregist_event = 2

cntl_error = 3

event_already_registered = 4

event_not_registered = 5.

Parameter	Bedeutung	Mögliche Werte
register	Registrierung des Ereignisses	true Ereignis registrieren (Defaultwert) false Ereignis nicht registrieren
appl_event	Auswahl zwischen System- und Anwendungsereignis	space Systemereignis (Defaultwert) 'x' Anwendungsereignis
navigate_on_dblick	lokale Navigation	false keine lokale Navigation (Defaultwert) true lokale Navigation

REGISTER_EVENT_F1

Verwendung

Mit dieser Methode wird das Event F1 (Funktionstaste) registriert.



Für ein transparentes Coding wird empfohlen, diese Methode direkt nach der Erzeugung der Instanz (`create object`) aufzurufen.

Funktionsumfang

call method textedit->register_event_f1

exporting

register = register

appl_event = appl_event

exceptions

error_regist_event = 1

error_unregist_event = 2

cntl_error = 3

event_already_registered = 4

event_not_registered = 5.

Parameter	Bedeutung	Mögliche Werte
register	Registrierung des Ereignisses	true Ereignis registrieren (Defaultwert) false Ereignis nicht registrieren
appl_event	Auswahl zwischen System- und Anwendungsereignis	space Systemereignis (Defaultwert) 'x' Anwendungsereignis

REGISTER_EVENT_F4

REGISTER_EVENT_F4

Verwendung

Mit dieser Methode wird das Event F4 (Funktionstaste) registriert.



Für ein transparentes Coding wird empfohlen, diese Methode direkt nach der Erzeugung der Instanz (`create object`) aufzurufen.

Funktionsumfang

call method textedit->register_event_f4

exporting

register = register

appl_event = appl_event

exceptions

error_regist_event = 1

error_unregist_event = 2

cntl_error = 3

event_already_registered = 4

event_not_registered = 5.

Parameter	Bedeutung	Mögliche Werte
register	Registrierung des Ereignisses	true Ereignis registrieren (Defaultwert) false Ereignis nicht registrieren
appl_event	Auswahl zwischen System- und Anwendungsereignis	space Systemereignis (Defaultwert) 'x' Anwendungsereignis

REGISTER_EVENT_FILEDROP

Verwendung

Mit dieser Methode wird das Event Filedrop registriert.



Für ein transparentes Coding wird empfohlen, diese Methode direkt nach der Erzeugung der Instanz (`create object`) aufzurufen.

Funktionsumfang

call method textedit->register_event_filedrop

exporting

register = register

appl_event = appl_event

exceptions

error_regist_event = 1

error_unregist_event = 2

cntl_error = 3

event_already_registered = 4

event_not_registered = 5.

Parameter	Bedeutung	Mögliche Werte
register	Registrierung des Ereignisses	true Ereignis registrieren (Defaultwert) false Ereignis nicht registrieren
appl_event	Auswahl zwischen System- und Anwendungsereignis	space Systemereignis (Defaultwert) 'x' Anwendungsereignis

REPLACE_ALL

REPLACE_ALL

Verwendung

Mit dieser Methode können Sie die mit SEARCH_STRING gefundenen Texte durch den in REPLACE_STRING angegebenen Text ersetzen.

Funktionsumfang

call method textedit->replace_all

exporting

case_sensitive_mode = case_sensitive_mode

replace_string = replace_string

search_string = search_string

whole_word_mode = whole_word_mode

changing

counter = counter

exceptions

error_cntl_call_method = 1


invalid_parameter = 2.

Parameter	Bedeutung	mögliche Werte
case_sensitive_mode	Groß- oder Kleinschreibung	false nicht beachten (Defaultwert) true beachten
replace_string	Text, der den gefundenen Text des SEARCH_STRING ersetzt	
search_string	zu ersetzender Text	
whole_word_mode	nur ganze Wörter sollen ersetzt werden	false nicht beachten (Defaultwert) true beachten
counter	Rückgabewert, der die Anzahl, wie oft der String ersetzt wurde, angibt	



Falls Sie beide Parameter SEARCH_STRING und REPLACE_STRING nicht gesetzt haben, erscheint das Dialogfenster *Suchen-Ersetzen*.

Es wird von der Position des Cursors bis an das Ende des Textes gesucht. Mit der Methode [SET_SELECTION_INDEXES \[Seite 70\]](#), [SET_SELECTION_POS \[Seite 71\]](#) oder [SET_SELECTION_POS_IN_LINE \[Seite 72\]](#) können Sie die Position des Cursors setzen. Falls alle Stellen innerhalb eines Textes ersetzt werden sollen, sollte die Selektion auf die erste Position im Text gesetzt werden.

 Das Ersetzen des Textes ist nicht auf den selektierten Bereich beschränkt.

SAVE_AS_LOCAL_FILE**SAVE_AS_LOCAL_FILE****Verwendung**

Mit dieser Methode können Sie Text in eine lokale Datei speichern.

Funktionsumfang

call method textedit->save_as_local_file

exporting
file_name = file_name

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung
file_name	Name der zu speichernden Datei. Eine Pfadangabe im UNC-Format ist möglich



Falls Sie den Parameter FILE_NAME nicht gesetzt haben, erscheint ein Dialogfenster zur Eingabe des Dateinamens und der Pfadangabe.

SELECT_LINES

Verwendung

Diese Methode ermöglicht es Ihnen, einen Zeilenbereich zu markieren. Standardmäßig wird der Zeilenbereich automatisch in den sichtbaren Bereich gesetzt.

Funktionsumfang

call method textedit->select_lines

exporting

from_line = from_line

to_line = to_line

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung
from_line	Startzeile des Zeilenbereiches
to_line	Endzeile (inklusive) des Zeilenbereiches

Falls das Setzen der Selektion für den Anwender nicht sichtbar erfolgen soll, gehen Sie folgendermaßen vor:

1. Holen Sie sich die augenblickliche Position des Cursors.
2. Schalten Sie das erneute Zeichnen mit der Methode [AUTO_REDRAW \[Seite 15\]](#), Parameter ENABLE_REDRAW = False, ab.
3. Setzen Sie die Selektion für den gewünschten Bereich und führen Sie gegebenenfalls weitere Aktionen durch.
4. Setzen Sie gegebenenfalls die neue Position.
5. Setzen Sie das automatische Verhalten des erneuten Zeichnens mit der Methode [AUTO_REDRAW \[Seite 15\]](#), Parameter ENABLE_REDRAW = True, wieder zurück.

SET_AUTOINDENT_MODE

SET_AUTOINDENT_MODE

Verwendung

Mit dieser Methode schalten Sie den automatischen Modus zur Einrückung von Zeilen an oder aus.

Weitere Informationen zur Einrückung von Zeilen finden Sie unter [INDENT_LINES \[Seite 43\]](#), [UNINDENT_LINES \[Seite 84\]](#), [INDENT_SELECTION \[Seite 44\]](#) und [UNINDENT_SELECTION \[Seite 85\]](#).

Funktionsumfang

call method textedit->set_autoindent_mode

exporting

auto_indent = auto_indent

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung	mögliche Werte
auto_indent	an- oder ausschalten des automatischen Modus für die Einrückung von Zeilen	true anschalten false ausschalten

SET_COMMENTS_STRING

Verwendung

Falls an der ersten Position in einer Zeile eine bestimmte Zeichenfolge steht, ermöglicht Ihnen diese Methode, diese Zeile automatisch als Kommentar zu kennzeichnen.



Beim ABAP Editor ist die Zeichenfolge zur Kennzeichnung einer Kommentarzeile der Stern *.

Die Hervorhebung der Kommentarzeile selbst wird durch Aufrufen der Methode [SET_HIGHLIGHT_COMMENTS_MODE \[Seite 64\]](#) an- oder ausgeschaltet.

Funktionsumfang

call method textedit->set_comments_string

exporting
comments_string = comments_string

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung
comments_string	Zeichenfolge

SET_DRAGDROP

SET_DRAGDROP

Verwendung

Mit dieser Methode setzen Sie das Verhalten beim Drag&Drop.

Funktionsumfang

call method textedit->set_dragdrop

```
exporting  
dragdrop = dragdrop.
```

Parameter	Bedeutung
dragdrop	Drag&Drop Objekt vom Typ CL_DRAGDROP. Dieses Objekt müssen Sie vorher erzeugt und entsprechend initialisiert haben. Weitere Informationen finden Sie unter Drag&Drop [Extern] .

SET_FILEDROP_MODE

Verwendung

Mit dieser Methode setzen Sie den Modus beim Drag&Drop von Dateien.



Den Modus können Sie bereits beim Erzeugen einer Instanz ([CONSTRUCTOR \[Seite 18\]](#)) setzen.

Funktionsumfang

call method textedit->set_filedrop_mode

exporting
filedrop_mode = filedrop_mode

exceptions
error_cntl_call_method = 1
invalid_parameter = 2.

Parameter	Bedeutung	mögliche Werte
filedrop_mode	Parameter, der das Verhalten beim Drag&Drop steuert	<p>dropfile_event_off Nur eine Datei, die den kompletten existierenden Text ersetzt, kann benutzt werden (Defaultwert).</p> <p>dropfile_event_single Nur eine Datei kann benutzt werden. Die Datei wird nicht in den Editor eingebunden, sondern das Datei-Drop-Ereignis wird ausgelöst. Der Pfad kann anhand der Methode GET_PATH_OF_DROPPED_FILES [Seite 29] angegeben werden.</p> <p>dropfile_event_multiple Mehrere Dateien können benutzt werden. Die Dateien werden nicht in den Editor eingebunden, sondern das Datei-Drop-Ereignis wird ausgelöst. Eine Liste der Pfade kann anhand der Methode GET_PATH_OF_DROPPED_FILES [Seite 29] angegeben werden.</p>

SET_FIRST_VISIBLE_LINE**SET_FIRST_VISIBLE_LINE****Verwendung**

Mit dieser Methode können Sie die angegebene Zeile als erste sichtbare im Editorfenster setzen.

Funktionsumfang

call method textedit->set_first_visible_line

exporting
line = line

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung
line	Zeilennummer

SET_HIGHLIGHT_BREAKPOINTS_MODE

Verwendung

Mit dieser Methode können Sie das Setzen eines Breakpoints an- oder ausschalten.



Diese Methode wird ausschließlich im ABAP Editor benötigt.

Funktionsumfang

call method textedit->set_highlight_breakpoints_mode

exporting

highlight_breakpoints_mode = highlight_breakpoints_mode

exceptions

error_cntl_call_method = 1

invalid_parameter = 2.

Parameter	Bedeutung	mögliche Werte
highlight_breakpoints_mode	an- oder ausschalten des Breakpoints	true anschalten (Defaultwert) false ausschalten



Falls dieser Modus angeschaltet ist, sind die Methoden zur expliziten Hervorhebung einzelner Textbereiche ([HIGHLIGHT_LINES \[Seite 41\]](#), [HIGHLIGHT_SELECTION \[Seite 42\]](#)) wirkungslos.

SET_HIGHLIGHT_COMMENTS_MODE

SET_HIGHLIGHT_COMMENTS_MODE**Verwendung**

Mit dieser Methode können Sie die automatische Hervorhebung von Kommentarzeilen an- oder ausschalten. Welche Zeilen als Kommentarzeilen erkannt werden, ist durch die Methode [SET_COMMENTS_STRING \[Seite 59\]](#) bestimmt.

Funktionsumfang

call method textedit->set_highlight_comments_mode

exporting

highlight_comments_mode = highlight_comments_mode

exceptions

error_cntl_call_method = 1

invalid_parameter = 2.

Parameter	Bedeutung	mögliche Werte
highlight_comments_mode	an- oder ausschalten der automatischen Hervorhebung	true anschalten (Defaultwert) false ausschalten



Falls dieser Modus angeschaltet ist, sind die Methoden zur expliziten Hervorhebung einzelner Textbereiche ([HIGHLIGHT_LINES \[Seite 41\]](#), [HIGHLIGHT_SELECTION \[Seite 42\]](#)) wirkungslos.

SET_LOCAL_CONTEXTMENU_MODE

Verwendung

Mit dieser Methode können Sie das lokale Kontext Menü sichtbar oder unsichtbar schalten.

Funktionsumfang

call method textedit->set_local_contextmenu_mode

exporting
visible = visible

exceptions
error_cntl_call_method = 1
invalid_parameter = 2.

Parameter	Bedeutung	Mögliche Werte
visible	Sichtbarkeit des lokalen Kontext Menüs	false Kontext Menü unsichtbar (Defaultwert) true Kontext Menü sichtbar

SET_NAVIGATE_ON_DBLCLICK

SET_NAVIGATE_ON_DBLCLICK**Verwendung**

Mit dieser Methode können Sie den Modus für die Reaktion auf einen Doppelklick setzen.

Funktionsumfang

call method textedit->set_navigate_on_dblick

```
exporting
  navigate_on_dblick_mode = navigate_on_dblick_mode
```

```
exceptions
  error_cntl_call_method = 1
  invalid_parameter = 2.
```

Parameter	Bedeutung	Mögliche Werte
navigate_on_dblick_mode	Modus für Reaktion auf einen Doppelklick	true Vorwärtsnavigation eingeschaltet (Defaultwert) false Vorwärtsnavigation ausgeschaltet

SET_READONLY_MODE

Verwendung

Diese Methode ermöglicht es Ihnen, zwischen dem Lesemodus und dem Eingabemodus des TextEdit Controls umzuschalten.

Beide Modi unterscheiden sich standardmäßig durch ihre Hintergrundfarben. Die Farben werden den allgemeinen Einstellungen des SAPGUIs entnommen.

Funktionsumfang

call method textedit->set_readonly_mode

exporting

read_only = read_only

exceptions

error_cntl_call_method = 1

invalid_parameter = 2.

Parameter	Bedeutung	mögliche Werte
read_only	Eingabemodus des TextEdit Controls	true keine Eingabe möglich (Defaultwert) false Eingabe möglich

SET_SELECTED_TEXT_AS_R3TABLE

SET_SELECTED_TEXT_AS_R3TABLE**Verwendung**

Die Methode ermöglicht es Ihnen, Text aus einer R/3-Tabelle einzufügen. Der Text wird an der Stelle des Cursors eingefügt. Eventuell selektierter Text wird überschrieben. Die Übergabe des Textes erfolgt ohne den Informationen über den Zeilenumbruch.

Funktionsumfang

call method textedit->set_selected_text_as_r3table

exporting

table = table

enable_editing_protected_text = enable_editing_protected_text

exceptions

error_dp = 1

error_db_create = 2.

Parameter	Bedeutung	Mögliche Werte
table	R/3-Tabelle mit Text	
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden



Falls Sie eine Tabelle verwenden, deren Zeilenlänge kleiner als die im Control dargestellte ist, so wird der Text lediglich bis zu dieser übergeben. Der restliche Text wird abgeschnitten und geht damit verloren.

Ist die Zeilenlänge der Tabelle größer, so werden Leerzeichen hinzugefügt.

SET_SELECTED_TEXT_AS_STREAM

Verwendung

Die Methode ermöglicht es Ihnen, Text aus einer R/3-Tabelle im Form eines Streams einzufügen. Der Text wird an der Stelle des Cursors eingefügt. Eventuell selektierter Text wird überschrieben. Die Übergabe des Textes erfolgt mit Informationen über den Zeilenumbruch,

Funktionsumfang

call method textedit->set_selected_text_as_stream

exporting

selected_text = selected_text

enable_editing_protected_text = enable_editing_protected_text

exceptions

error_dp = 1

error_db_create = 2.

Parameter	Bedeutung	Mögliche Werte
selected_text	R/3-Tabelle mit Text	
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	<p>false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert)</p> <p>true geschützter Textbereich kann bearbeitet werden</p>

SET_SELECTION_INDEXES

SET_SELECTION_INDEXES

Verwendung

Mit dieser Methode können Sie eine Selektion für einen Textbereich setzen (abhängig von einem Zeichenindex). Standardmäßig wird diese Position automatisch in den sichtbaren Bereich gesetzt.

Funktionsumfang

call method textedit->set_selection_indexes

exporting

from_index = from_index

to_index = to_index

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung
from_index	Zeichenindex, von dem an die Selektion beginnen soll. Die Normierung erfolgt auf das erste Zeichen im Control.
to_index	Zeichenindex, an dem die Selektion enden soll. Die Normierung erfolgt auf das erste Zeichen im Control.



Falls beide Parameter übereinstimmen, wird kein Text selektiert, sondern nur der Cursor an die entsprechende Position gesetzt.

Falls das Setzen der Selektion für den Anwender nicht sichtbar erfolgen soll, gehen Sie folgendermaßen vor:

1. Holen Sie sich die augenblickliche Position des Cursors.
2. Schalten Sie das erneute Zeichnen mit der Methode [AUTO_REDRAW \[Seite 15\]](#), Parameter ENABLE_REDRAW = False, ab.
3. Setzen Sie die Selektion für den gewünschten Bereich und führen Sie gegebenenfalls weitere Aktionen durch.
4. Setzen Sie gegebenenfalls die alte Position.
5. Setzen Sie das automatische Verhalten des neu Zeichnen mit der Methode [AUTO_REDRAW \[Seite 15\]](#), Parameter ENABLE_REDRAW = True, wieder zurück.

SET_SELECTION_POS

Verwendung

Mit dieser Methode können Sie eine Selektion für einen Textbereich setzen. Standardmäßig wird diese Position automatisch in den sichtbaren Bereich gesetzt.

Funktionsumfang

call method textedit->set_selection_pos

```
exporting
  from_line = from_line
  from_pos = from_pos
  to_line = to_line
  to_pos = to_pos
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung
from_line	Zeilennummer, in der die Selektion beginnen soll
from_pos	Position innerhalb der Zeile, an der die Selektion beginnen soll
to_line	Zeilennummer, in der die Selektion enden soll
to_pos	Position innerhalb der Zeile, an der die Selektion enden soll



Falls die Zeilen- und Positionsnummern übereinstimmen, wird kein Text selektiert, sondern nur der Cursor an die entsprechende Position gesetzt.

Falls das Setzen der Selektion für den Anwender nicht sichtbar erfolgen soll, gehen Sie folgendermaßen vor:

6. Holen Sie sich die augenblickliche Position des Cursors.
7. Schalten Sie das erneute Zeichnen mit der Methode [AUTO_REDRAW \[Extern\]](#), Parameter ENABLE_REDRAW = False, ab.
8. Setzen Sie die Selektion für den gewünschten Bereich und führen Sie gegebenenfalls weitere Aktionen durch.
9. Setzen Sie gegebenenfalls die alte Position.
10. Setzen Sie das automatische Verhalten des neu Zeichnen mit der Methode [AUTO_REDRAW \[Extern\]](#), Parameter ENABLE_REDRAW = True, wieder zurück.

SET_SELECTION_POS_IN_LINE**SET_SELECTION_POS_IN_LINE****Verwendung**

Diese Methode ermöglicht es Ihnen, den Cursor innerhalb einer Zeile an eine Position zu setzen.

Funktionsumfang

call method textedit->set_selection_pos_in_line

exporting
line = line
pos = pos

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung
line	Zeilennummer, in der die Selektion beginnen soll
pos	Position innerhalb der Zeile

SET_SPACES_ON_INDENT

Verwendung

Mit dieser Methode setzen Sie die Anzahl der Leerzeichen für eine Einrückung von Zeilen oder von markiertem Textbereich. Weitere Informationen finden Sie unter [INDENT_LINES \[Seite 43\]](#), [INDENT_SELECTION \[Seite 44\]](#), [UNINDENT_LINES \[Seite 84\]](#) und [UNINDENT_SELECTION \[Seite 85\]](#). Die Anzahl der Leerzeichen wird der Klassenkonstanten M_SPACES_ON_INDENT übergeben (Weitere Informationen finden Sie unter [Klassenkonstanten \[Seite 97\]](#)).

Funktionsumfang

call method textedit->set_spaces_on_indent

exporting

number_of_spaces = number_of_spaces

exceptions

error_cntl_call_method = 1

invalid_parameter = 2.

Parameter	Bedeutung
number_of_spaces	Anzahl der Leerzeichen

SET_STATUSBAR_MODE

SET_STATUSBAR_MODE

Verwendung

Mit dieser Methode können Sie die Sichtbarkeit der Statuszeile an- oder ausschalten.

Funktionsumfang

call method textedit->set_statusbar_mode

```
exporting  
  statusbar_mode = statusbar_mode
```

```
exceptions  
  error_cntl_call_method = 1  
  invalid_parameter = 2.
```

Parameter	Bedeutung	mögliche Werte
statusbar_mode	Modus der Statuszeile	false Statuszeile unsichtbar (Defaultwert) true Statuszeile sichtbar

SET_STATUS_TEXT

Verwendung

Mit dieser Methode können Sie in der Statuszeile des Controls einen Text anzeigen. Die Sichtbarkeit der Statuszeile können Sie durch die Methode [SET_STATUSBAR_MODE \[Seite 74\]](#) an- oder ausschalten.

Funktionsumfang

call method textedit->set_status_text

exporting

status_text = status_text

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung
status_text	Text, der in der Statuszeile angezeigt werden soll

SET_TEXTMODIFIED_STATUS

SET_TEXTMODIFIED_STATUS

Verwendung

Mit dieser Methode setzen Sie den Änderungsstatus des Textes. Die Methode [GET_TEXTMODIFIED_STATUS \[Seite 34\]](#) liefert Ihnen den Änderungsstatus eines Textes.



Falls Sie Text mit den Methoden [SET_TEXT_AS_R3TABLE \[Seite 77\]](#) und [SET_TEXT_AS_STREAM \[Seite 78\]](#) holen, wird der Änderungsstatus nicht zurückgesetzt.



Diese Methode ist für das absichtliche Übersehen von Textänderungen nützlich. Falls Sie z.B. den Text vom Control erhalten und ihn permanent in der Datenbank speichern, können Sie ihn auf nicht geändert setzen, ohne Parametern anzugeben.

Funktionsumfang

call method textedit->set_textmodified_status

exporting
status = status

exceptions
error_cntl_call_method = 1.

Parameter	Bedeutung	Mögliche Werte
status	Änderungsstatus	false Text nicht geändert (Defaultwert) true Text geändert

SET_TEXT_AS_R3TABLE

Verwendung

Mit dieser Methode können Sie Text aus einer R/3-Tabelle setzen. Bereits vorhandener Text wird überschrieben. Die Übergabe des Textes erfolgt ohne den Informationen über den Zeilenumbruch.

Funktionsumfang

call method textedit->set_text_as_r3table

exporting
table = table

exceptions
error_dp = 1
error_dp_create = 2.

Parameter	Bedeutung
table	R/3-Tabelle mit Text



Falls Sie eine Tabelle verwenden, deren Zeilenlänge kleiner als die im Control dargestellte ist, so wird der Text lediglich bis zu dieser übergeben. Der restliche Text wird abgeschnitten und geht damit verloren.

Ist die Zeilenlänge der Tabelle größer, so werden Leerzeichen hinzugefügt.

SET_TEXT_AS_STREAM**SET_TEXT_AS_STREAM****Verwendung**

Mit dieser Methode können Sie Text aus einer R/3-Tabelle im Form eines Streams setzen. Bereits vorhandener Text wird überschrieben. Die Übergabe des Textes erfolgt mit Informationen über den Zeilenumbruch,

Funktionsumfang

call method textedit->set_text_as_stream

exporting
text = text

exceptions
error_dp = 1
error_dp_create = 2.

Parameter	Bedeutung
text	R/3-Tabelle mit Text

SET_TOOLBAR_MODE

Verwendung

Mit dieser Methode können Sie die Sichtbarkeit der Toolbar an- oder ausschalten.

Funktionsumfang

call method textedit->set_toolbar_mode

exporting
 toolbar_mode = toolbar_mode

exceptions
 error_cntl_call_method = 1
 invalid_parameter = 2.

Parameter	Bedeutung	mögliche Werte
toolbar_mode	Modus der Toolbar	false Toolbar unsichtbar (Defaultwert) true Toolbar sichtbar

SET_WORDBREAK_PROCEDURE

SET_WORDBREAK_PROCEDURE

Verwendung

Mit dieser Methode können Sie das Verhalten des Zeilenumbruchs bei normalen Text und ABAP Coding bestimmen.

Funktionsumfang

call method textedit->set_wordbreak_procedure

exporting

text_type = text_type

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung	Mögliche Werte
text_type	Verhalten des Zeilenumbruchs	<code>text_standard</code> Text <code>text_abap</code> ABAP Coding

SET_WORDWRAP_BEHAVIOR

Verwendung

Mit dieser Methode können Sie das Verhalten des Zeilenumbruches setzen.

Funktionsumfang

call method textedit->set_wordwrap_behavior

```
exporting
  wordwrap_mode = wordwrap_mode
  wordwrap_position = wordwrap_position
  wordwrap_to_linebreak_mode = wordwrap_to_linebreak_mode
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
wordwrap_mode	Verhalten des Zeilenumbruchs	wordwrap_off kein Zeilenumbruch wordwrap_at_windowborder Zeilenumbruch an der Fenstergrenze (Defaultwert) wordwrap_at_fixed_position Zeilenumbruch an einer festen Position Weitere Informationen finden Sie unter Klassenkonstanten [Seite 97] und der Methode CONSTRUCTOR [Seite 18] .
wordwrap_position	Falls wordwrap_mode = textedit->wordwrap_at_fixed_position , Position für den automatischen Zeilenumbruch in einer Zeile	Defaultwert: -1
wordwrap_to_linebreak_mode	Umwandlung der weichen in harte Zeilenumbrüche beim Speichern in das R/3-System	false weiche Zeilenumbrüche werden beim Speichern ignoriert true Umwandlung von weiche in harte Zeilenumbrüche beim Speichern

UNCOMMENT_LINES

UNCOMMENT_LINES

Verwendung

Mit dieser Methode können Sie eine Anzahl von Kommentarzeilen in normale Zeilen umwandeln.

Funktionsumfang

call method textedit->uncomment_lines

```
exporting
  from_line = from_line
  to_line = to_line
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
from_line	Beginn der Zeilen	
to_line	Ende der Zeilen	
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

UNCOMMENT_SELECTION

Verwendung

Mit dieser Methode können Sie markierte Kommentarzeilen in normale Zeilen umwandeln.

Funktionsumfang

call method textedit->uncomment_selection

exporting

enable_editing_protected_text = enable_editing_protected_text

exceptions

error_cntl_call_method = 1.

Parameter	Bedeutung	Mögliche Werte
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	<p>false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert)</p> <p>true geschützter Textbereich kann bearbeitet werden</p>

UNINDENT_LINES

UNINDENT_LINES

Verwendung

Diese Methode verringert die Einrückung eines Zeilenbereiches um eine Anzahl von Leerzeichen. Die Anzahl setzen Sie mit der Methode [SET_SPACES_ON_INDENT \[Seite 73\]](#) und den Parameter M_SAPCES_ON_INDENT. Die Verringerung erfolgt nur, falls Leerzeichen am Beginn des Zeilenbereiches vorhanden sind.

Funktionsumfang

call method textedit->unindent_lines

```
exporting
  from_line = from_line
  to_line = to_line
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
from_line	Beginn des Zeilenbereiches, der eingerückt werden soll	
to_line	Ende des Zeilenbereiches, der eingerückt werden soll	
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

UNINDENT_SELECTION

Verwendung

Diese Methode verringert die Einrückung eines markierten Textbereiches um eine Anzahl von Leerzeichen. Die Anzahl setzen Sie mit der Methode [SET_SPACES_ON_INDENT \[Seite 73\]](#) und den Parameter M_SAPCES_ON_INDENT. Die Verringerung erfolgt nur, falls Leerzeichen am Beginn des Textbereiches vorhanden sind.

Funktionsumfang

call method textedit->unindent_selection

```
exporting
  enable_editing_protected_text = enable_editing_protected_text
```

```
exceptions
  error_cntl_call_method = 1.
```

Parameter	Bedeutung	Mögliche Werte
enable_editing_protected_text	Bearbeiten von geschützten Textbereichen	false geschützter Textbereich kann nicht bearbeitet werden (Defaultwert) true geschützter Textbereich kann bearbeitet werden

funktionale Sortierung

funktionale Sortierung

Sie haben die Hilfe zur Anwendung aufgerufen. Zu dem aktuellen R/3-Kontext sind folgende Informationen verfügbar:

Die funktionale Sortierung ermöglicht es Ihnen, Funktionen des TextEdit Controls zielgerichtet zu programmieren. Die einzelnen Funktionen sind in Themengebiete eingeteilt. Innerhalb dieser finden Sie dann die relevanten Methoden alphabetisch sortiert.

Themengebiete

[Control anlegen und abbauen \[Extern\]](#)

[Setzen und Holen von Text \[Extern\]](#)

[Erfragen und Setzen von Textpositionen \[Extern\]](#)

[Hervorheben und Schützen von Textteilen \[Extern\]](#)

[Suchen und Ersetzen von Textteilen \[Extern\]](#)

[Statusbar \[Extern\]](#)

[Toolbar \[Extern\]](#)

[Sonstige \[Extern\]](#)

Control anlegen

[CONSTRUCTOR \[Seite 18\]](#)

Setzen und Holen von Text

Setzen und Holen von Text

Es sind zwei unterschiedliche Lösungen zum Datenaustausch mit Tabellen implementiert. Zum einen ein Datenaustausch mit der Information über den Zeilenumbruch und zum anderen ohne. Dies spiegelt sich in dem Satz von Methoden mit der Endung AS_STREAM (mit den Informationen über den Zeilenumbruch) und AS_R3TABLE (ohne die Informationen über den Zeilenumbruch) wieder.

Holen von Text	Setzen von Text
GET_LINE_TEXT [Seite 28]	
GET_SELECTED_TEXT_AS_R3TABLE [Seite 30]	SET_SELECTED_TEXT_AS_R3TABLE [Seite 68]
GET_SELECTED_TEXT_AS_STREAM [Seite 31]	SET_SELECTED_TEXT_AS_STREAM [Seite 69]
GET_TEXT_AS_R3TABLE [Seite 35]	SET_TEXT_AS_R3TABLE [Seite 77]
GET_TEXT_AS_STREAM [Seite 37]	SET_TEXT_AS_STREAM [Seite 78]
OPEN_LOCAL_FILE [Seite 46]	SAVE_AS_LOCAL_FILE [Seite 56]

Löschen von Text	
DELETE_TEXT [Seite 21]	



Falls Sie die GET-Methoden vom Typ AS_R3TABLE benutzen und falls Sie eine Tabelle verwenden, deren Zeilenlänge kleiner als die im Control dargestellte ist, so wird der Text lediglich bis zu dieser übergeben. Der restliche Text wird abgeschnitten und geht damit verloren.

Ist die Zeilenlänge der Tabelle größer, so werden Leerzeichen hinzugefügt.

Erfragen und Setzen von Textpositionen

Generell sind zwei unterschiedliche Arten des Zugriffs auf Textpositionen implementiert. Zum einen der Zugriff über Zeilen und Positionen innerhalb einer Zeile und zum anderen über den absoluten Zeichenindex. Zeichenindex bedeutet, daß alle Zeichen im Text laufend durchnummeriert sind.

Der wesentliche Unterschied besteht in der Visualisierung der per Tabelle an das Control gesendeten Daten. Bei eingestelltem automatischen Zeilenumbruch entsprechen die Zeilennummern in der Tabelle nicht notwendigerweise den visualisierten Zeilennummern im Control. Dies gilt natürlich analog auch für die Positionen innerhalb einer Zeile. Das Verhalten beim Zeilenumbruch legen Sie im Konstruktor fest (weitere Informationen finden Sie unter [CONSTRUCTOR \[Seite 18\]](#)).

Besonders anschaulich ist dieser Unterschied beim eingestellten Verhalten des Zeilenumbruches WORDWRAP_AT_WINDOWBORDER (automatischer Umbruch an der Fenstergrenze des TextEdit). Falls die Größe des SAPGUI-Fensters verändert wird, verändert sich, sofern ein entsprechendes Resize-Verhalten des Containers aktiv ist, auch die Größe des TextEdit. Dies kann beim Verkleinern des SAPGUI-Fensters zu einer kleineren Breite des TextEdit führen. Zeilen mit Wörtern, die an die Fenstergrenze stoßen, werden dann automatisch umgebrochen, wodurch neue Zeilen entstehen. Obwohl im TextEdit keine Eingabe erfolgt ist, entsprechen somit die Zeilen und Positionen innerhalb einer Zeile nicht mehr denjenigen der am Backend gehaltenen internen Tabelle.

Analog ist ein Beispiel für das Verhalten des Zeilenumbruches WORDWRAP_AT_FIXED_POSITION aufbaubar.



Möchten Sie also von der Positionen aus der internen Tabelle heraus direkt auf Positionen im TextEdit schließen, ist im Fall der oben erwähnten Verhaltens nur ein Zugriff über den Zeichenindex sinnvoll.

Zugriff über Zeilen- und Positionsnummern

Zeilen und Positionen innerhalb einer Zeile werden mit eins beginnend numeriert, d.h. Zeile 1 Position 1 ist die erste Position im Text.

Zeilen- und Positionsnummern, die sich außerhalb des Textbereiches befinden, werden einem logischen Wert zugeordnet. Ist eine Nummer kleiner als 1, wird sie auf 1 gesetzt. Ist sie größer als die vorhandenen Nummern von Zeilen oder Positionen, wird sie auf die Nummer der letzten verfügbaren Zeile oder Position gesetzt.

Erfragen von Textpositionen	Setzen von Textpositionen
GET_FIRST_VISIBLE_LINE [Seite 27]	SET_FIRST_VISIBLE_LINE [Seite 62]
GET_SELECTION_POS [Seite 33]	SET_SELECTION_POS [Seite 71]
GO_TO_LINE [Seite 39]	
	SELECT_LINES [Seite 57]
	SET_SELECTION_POS_IN_LINE [Seite 72]

Erfragen und Setzen von Textpositionen**Zugriff über den Zeichenindex**

Der Zeichenindex wird mit 0 beginnend numeriert. Die Position 0 kennzeichnet die Position vor dem ersten Zeichen. Die Information über die Textposition ist unabhängig vom Verhalten des Zeilenumbruches (siehe oben).

Erfragen von Textpositionen	Setzen von Textpositionen
GET_SELECTION_INDEXES [Seite 32]	SET_SELECTION_INDEXES [Seite 70]

Hervorheben und Schützen von Textteilen

Erneutes Zeichnen des Controls[AUTO_REDRAW \[Seite 15\]](#)**Hervorhebung von Textteilen ein- und abschalten**[HIGHLIGHT_LINES \[Seite 41\]](#)[HIGHLIGHT_SELECTION \[Seite 42\]](#)**Schutz gegen Eingabe ein- und abschalten**[PROTECT_LINES \[Seite 47\]](#)[PROTECT_SELECTION \[Seite 48\]](#)**Kommentarzeilen**[SET_COMMENTS_STRING \[Seite 59\]](#)[SET_HIGHLIGHT_COMMENTS_MODE \[Seite 64\]](#)**Umschalten zwischen Eingabe- und Lesemodus**[SET_READONLY_MODE \[Seite 67\]](#)**Verschieben in den sichtbaren Bereich des Editorfensters**[MAKE_SELECTION_VISIBLE \[Seite 45\]](#)

Suchen und Ersetzen von Textteilen

Suchen und Ersetzen von Textteilen
EMPTY_UNDO_BUFFER [Seite 22]
FIND_AND_REPLACE [Seite 23]
FIND_AND_SELECT_TEXT [Seite 25]
REPLACE_ALL [Seite 54]

Statusbar

Setzen der Statusbar	ReadOnly-Instanz-Attribute
SET_STATUSBAR_MODE [Seite 74]	Instanz-Attribute [Seite 98]
SET_STATUS_TEXT [Seite 75]	

Toolbar

Toolbar

Setzen der Toolbar	ReadOnly-Instanz-Attribute
SET_TOOLBAR_MODE [Seite 79]	Instanz-Attribute [Seite 98]

Sonstige


Setzen des Zeilenumbruches	
SET_WORDWRAP_BEHAVIOR [Seite 81]	

Ereignisse


Ereignisse

Wenn Sie auf Ereignisse in Ihrem ABAP-Programm reagieren möchten, müssen Sie sich auf diese Ereignisse registrieren. Dazu verwenden Sie die Methoden [REGISTER_EVENT_CONTEXT_MENU \[Seite 49\]](#), [REGISTER_EVENT_DBLCLICK \[Seite 50\]](#), [REGISTER_EVENT_F1 \[Seite 51\]](#), [REGISTER_EVENT_F4 \[Seite 52\]](#), [REGISTER_EVENT_FILEDROP \[Seite 53\]](#). Ereignisse auf die Sie sich nicht registriert werden, werden schon am Frontend gefiltert und gelangen nicht zum Backend. Weitere Informationen finden Sie unter [Ereignisbehandlung \[Extern\]](#).

Ereignis	Bedeutung / Verwendung
CONTEXT_MENU	Aufruf des Kontext Menüs über rechte Maustaste Weitere Informationen finden Sie unter Kontext Menü [Extern] .
CONTEXT_MENU_SELECTED	Auswahl eines Menüs im Kontext Menü Weitere Informationen finden Sie unter Kontext Menü [Extern] .
DBLCLICK	Doppelklick
F1	Drücken der Funktionstaste F1
F4	Drücken der Funktionstaste F4
FILEDROP ON_DRAG ON_DROP ON_DROP_COMPLETE ON_GET_FLAVOR	Ereignisse des Drag&Drop Weitere Informationen finden Sie unter Drag&Drop [Extern] .

 Beispiele zur Ereignisbehandlung finden Sie im Programm SAPTEXTEDIT_TEST_EVENTS in der Entwicklungsklasse SAPTEXTEDIT.

Klassenkonstanten

Klassenkonstante	Type / Value	Beschreibung / Verwendung
ABAP_COMMETNLINE_IDENTIFI ER	TYPE C VALUE *	Hervorheben von Kommentarzeilen bei der Verwendung des Controls als Quelltexteditor
BOOL_INITIAL FALSE TRUE	TYPE I VALUE -1 TYPE I VALUE 0 TYPE I VALUE 1	Nachbildung des Typen Bool
STRING_LENGTH	TYPE I VALUE 256	Typisierung von Strings. Bei der Übergabe von Parametern an das Control entspricht diese Definition der maximal übertragenen Menge von Zeichen zwischen Frontend und Backend.  Falls ein größerer String übergeben wird, erfolgt keine gesonderte Warnung.
WORDWRAP_OFF	TYPE I VALUE 0	kein automatisches Umbrechen des Textes im Control. Diese Definition benötigen Sie nur beim Kreieren eines Objektes im Konstruktor
WORDWRAP_AT_WINDOWBORD ER	TYPE I VALUE 1	automatisches Umbrechen des Textes an der Fenstergrenze des Controls. Diese Definition benötigen Sie nur beim Kreieren eines Objektes im Konstruktor
WORDWRAP_AT_FIXED_POSITI ON	TYPE I VALUE 2	automatisches Umbrechen des Textes an einer festen Position im Control. Diese Definition benötigen Sie nur beim Kreieren eines Objektes im Konstruktor



In der ABAP Programmiersprache gibt es keine Konstanten des Typen Bool. Zur Nachbildung dieses Typs sind die Integer-Konstanten `false` und `true` definiert. Diese Klassenkonstanten sind über `c1_gui_textedit=>>true` oder `c1_gui_textedit=>>false` ansprechbar.

Instanz-Attribute

Instanz-Attribute

Die folgenden Instanz-Attribute geben aktuelle Eigenschaften des Controls an. Sie besitzen alle das Flag "Read-Only". Read-Only bedeutet, daß diese Attribute lediglich benutzt, aber nicht verändert werden können.

Die Eigenschaften am Frontend werden nicht innerhalb des Controls gesetzt und können daher nur mit Set-Methoden aus dem ABAP-Proxy heraus geändert werden. Dies ermöglicht die Werte der Frontend-Eigenschaften auf der ABAP Seite mit Attributen wiederzugeben. Das Setzen erfordert nun den Aufruf der entsprechenden Set-Methode, die implizit das entsprechende Instanz-Attribut aktualisiert. Damit werden die Aufrufe der Get-Methoden umgangen und somit performance-teuere Flushes vermieden.


Die Initialisierung der Instanz-Attribute erfolgt erstmalig im Konstruktor ([CONSTRUCTOR \[Seite 18\]](#)) und wird bei erfolgreicher Ausführung der entsprechenden Set-Methoden aktualisiert.

Instanz-Attribut	Type / Value	Beschreibung / Verwendung
M_AUTOREDRAW_REFCONTAINER	TYPE I VALUE 0 READ-ONLY	Referenzzähler zur Steuerung des erneuten Zeichnens des Controls
M_AUTO_INDENT	TYPE I READ-ONLY	Wert für automatische Zeileneinrückung
M_COMMENTS_STRING(STRING_LENGTH)	TYPE C READ-ONLY	Zeichen, die am Anfang einer Zeile die ganze Zeile als Kommentar kennzeichnen. Der Wert wird durch die Methode SET_COMMENTS_STRING [Seite 59] gesetzt. Das Hervorheben selbst durch die Methode SET_HIGHLIGHT_COMMENTS_MODE [Seite 64] .
M_FILEDROP_MODE	TYPE I VALUE DROPFILE_EVENT_OFF READ-ONLY	Modus der Dateien beim Drag&Drop. Das Setzen erfolgt mit der Methode SET_FILEDROP_MODE [Seite 61] .
M_HIGHLIGHT_BREAKPOINTS_MODE	TYPE I READ-ONLY	Hervorhebung Breakpoint an- oder abgeschaltet (True oder False). Das Setzen erfolgt mit der Methode SET_HIGHLIGHT_BREAKPOINTS_MODE [Seite 63] .

Instanz-Attribute

M_HIGHLIGHT_COMMENTS_MODE	TYPE I READ-ONLY	Hervorhebung der Kommentarzeile an- oder abgeschaltet (True oder False). Das Setzen erfolgt durch die Methode SET_HIGHLIGHT_COMMENTS_MODE [Seite 64].
M_LOCAL_CONTEXTMENU_MODE	TYPE I READ-ONLY	Sichtbarkeit des Kontext Menüs (True: sichtbar, False unsichtbar). Das Setzen erfolgt durch die Methode SET_LOCAL_CONTEXTMENU_MODE [Seite 65].
M_READONLY_MODE	TYPE I READ-ONLY	Darstellung des Textes im geschützten oder eingabebereiten Modus (False oder True). Das Setzen erfolgt durch die Methode SET_READONLY_MODE [Seite 67].
M_SPACES_ON_INDENT	TYPE I VALUE 2 READ-ONLY	Anzahl der Leerzeichen, die beim Einrücken eingefügt bzw. gelöscht werden. Das Setzen erfolgt mit der Methode SET_SPACES_ON_INDENT [Seite 73].
M_STATUS_TEXT(StringLength)	TYPE C READ-ONLY	Text im ersten Element der Statuszeile. Das Setzen erfolgt durch die Methode SET_STATUS_TEXT [Seite 75].
M_STATUSBAR_MODE	TYPE I READ-ONLY	Anzeigen oder Nichtanzeigen der Statuszeile (False oder True). Das Setzen erfolgt durch die Methode SET_STATUSBAR_MODE [Seite 74].
M_TOOLBAR_MODE	TYPE I READ-ONLY	Anzeigen oder Nichtanzeigen der Toolbar (False oder True). Das Setzen erfolgt durch die Methode SET_TOOLBAR_MODE [Seite 79].
M_WORDBREAK_PROCEDURE	TYPE I READ-ONLY	Verhalten des Zeilenumbruchs bei normalen Text oder ABAP Coding (TEXT_STANDARD und TEXT_ABAP).

Instanz-Attribute

M_WORDWRAP_MODE	TYPE I READ-ONLY	<p>Verhalten des Zeilenumbruchs:</p> <ul style="list-style-type: none"> • Kein Zeilenumbruch (WORDWRAP_OFF) • Zeilenumbruch an der Fenstergrenze (WORDWRAP_AT_WINDOWBORDER) • Zeilenumbruch an einer festen Position (WORDWRAP_AT_FIXED_POSITION) <p>Das Setzen dieser Werte erfolgt beim Aufruf des Konstruktors (CONSTRUCTOR [Seite 18]) oder mit der Methode SET_WORDWRAP_BEHAVIOR [Seite 81].</p>
M_WORDWRAP_POSITION	TYPE I READ-ONLY	<p>Position in einer Zeile, an der ein Zeilenumbruch automatisch stattfinden soll. Das Setzen des Wertes erfolgt beim Aufruf des Konstruktors (CONSTRUCTOR [Seite 18]).</p> <p style="text-align: center;"></p> <p>Dieser Wert ist nur sinnvoll, falls M_WORDWRAP_MODE auf 2 bzw. WORDWRAP_AT_FIXED_POSITION gesetzt ist</p>

Instanz-Attribute

<p>M_WORDWRAP_TO_LINEBREAK REAK_MODE</p>	<p>TYPE I READ-ONLY</p>	<p>Umwandlung von weichen in harte Zeilenumbrüche beim Speichern in das R/3-System:</p> <ul style="list-style-type: none"> • Weiche Umbrüche werden beim Speichern ignoriert. Eine Zeile endet mit dem nächsten harten Umbruch (False) • Umwandlung von weichen in harte Umbrüche erfolgt beim Speichern (True) <p>Das Setzen des Wertes erfolgt beim Aufruf des Konstruktors (CONSTRUCTOR [Seite 18]).</p>
<p>M_NAVIGATE_ON_DBLCLICK</p>	<p>TYPE I READ-ONLY</p>	<p>Modus der Reaktion auf einen Doppelklick. Das Setzen des Wertes erfolgt mit der Methode SET_NAVIGATE_ON_DBLCLICK [Seite 66].</p>

Tastatur- und Maussteuerung im Editorfenster

Tastatur- und Maussteuerung im Editorfenster**Einfüge- und Überschreibmodus**

Tastaturkommando	Beschreibung
Einfg	Der Editor schaltet zwischen Einfüge- auf Überschreibmodus um

Clipboard-Steuerung

Tastaturkommando	Beschreibung
Umsch+Einfg Umsch+Strg+Einfg Strg+C Umsch+Strg+C	Einfügen des selektierten Textes in die Zwischenablage (Copy).
Umsch+Entf Umsch+Strg+Entf Strg+X Umsch+Strg+X	Ausschneiden von selektiertem Text (Cut). Dieser wird in das Clipboard eingefügt.
Umsch+Einfg Umsch+Strg+Einfg Strg+V Umsch+Strg+V	Einfügen des Clipboard-Textes an der Stelle des blinkenden Cursors (Paste).

Undo- / Redo-Steuerung

Tastaturkommando	Beschreibung
Strg+Z Alt+Rück	Undo
Strg+Y	Redo

Navigation im Text

Tastaturkommando	Beschreibung
Wählen Sie ↑	Der Cursor bewegt sich eine Zeile nach oben
↓	Der Cursor bewegt sich eine Zeile nach unten
←	Der Cursor bewegt sich ein Zeichen nach links
→	Der Cursor bewegt sich ein Zeichen nach rechts
Bild hoch	Der Cursor bewegt sich eine Seite nach oben
Bild unten	Der Cursor bewegt sich eine Seite nach unten
Pos1	Der Cursor befindet sich am Anfang der Zeile

Tastatur- und Maussteuerung im Editorfenster

Ende	Der Cursor befindet sich am Ende der Zeile
Strg+↑	Der Cursor bewegt sich zum Anfang des aktuellen Absatzes
Strg+↓	Der Cursor bewegt sich zum Anfang des folgenden Absatzes
Strg+←	Der Cursor bewegt sich zum nächsten linken Wortanfang
Strg+→	Der Cursor bewegt sich zum nächsten rechten Wortanfang
Strg+Bild hoch	Der Cursor bewegt sich vor das erste voll sichtbare Zeichen des Editorfensters
Strg+Bild unten	Der Cursor bewegt sich hinter das letzte voll sichtbare Zeichen des Editorfensters
Strg+Pos1	Der Cursor bewegt sich vor das erste Zeichen des gesamten Textes
Strg+Ende	Der Cursor bewegt sich hinter das letzte Zeichen des gesamten Textes

Selektion von Texten

Tastaturkommando	Beschreibung
Strg+A Umsch+Strg+A	Der ganze Text wird selektiert
Umsch+↑	Der Text, ab der aktuellen Position des Cursors eine ganze Zeile nach oben, wird selektiert
Umsch+↓	Der Text, ab der aktuellen Position des Cursors eine ganze Zeile nach unten, wird selektiert
Umsch+←	Der Text, ab der aktuellen Position des Cursors ein Zeichen nach links, wird selektiert
Umsch+→	Der Text, ab der aktuellen Position des Cursors ein Zeichen nach rechts, wird selektiert
Umsch+Bild hoch.	Der Text ab, der aktuellen Position des Cursors eine ganze sichtbare Editorseite nach oben, wird selektiert
Umsch+Bild unten	Der Text, ab der aktuellen Position des Cursors eine ganze sichtbare Editorseite nach unten, wird selektiert
Umsch+Pos1	Der Text, ab der aktuellen Position des Cursors bis zum Zeilenanfang, wird selektiert
Umsch+Ende	Der Text, ab der aktuellen Position des Cursors bis zum Zeilenende, wird selektiert
Umsch+Strg+↑	Der Text, ab der aktuellen Position des Cursors bis zum Anfang des aktuellen Absatzes, wird selektiert

Tastatur- und Maussteuerung im Editorfenster

Umsch+Strg+↓	Der Text, ab der aktuellen Position des Cursors bis zum Ende des aktuellen Absatzes, wird selektiert
Umsch+Strg+←	Der Text, ab der aktuellen Position des Cursors bis zum nächsten Wortanfang nach links, wird selektiert
Umsch+Strg+→	Der Text ab der aktuellen Position des Cursors bis zum nächsten Wortanfang nach rechts wird selektiert
Umsch+Strg+Bild hoch	Der Text, ab der aktuellen Position des Cursors bis zur ersten voll sichtbaren Zeile des Editorfensters, wird selektiert
Umsch+Strg+Bild unten	Der Text, ab der aktuellen Position des Cursors bis zur letzten voll sichtbaren Zeile des Editorfensters, wird selektiert
Umsch+Strg+Pos1	Der Text, ab der aktuellen Position des Cursors bis zum Anfang des gesamten Textes, wird selektiert
Umsch+Strg+Ende	Der Text, ab der aktuellen Position des Cursors bis zum Ende des gesamten Textes, wird selektiert
Linke Maustaste gedrückt ziehen	Der Textblock wird selektiert
Linke Mause am linken Rand einer Zeile einmalig drücken	Die gesamte Zeile wird selektiert
Linke Mause am linken Rand einer Zeile drücken und auf diesem nach oben oder unten ziehen	Mehrere aufeinanderfolgende Zeilen werden selektiert
Strg+Linke Mause am linken Rand einer Zeile einmalig drücken	Der gesamten Text wird selektiert

Suchen und Ersetzen

Ist ein Textbereich selektiert, so beziehen sich die Aktionen auf den selektierten Textbereich. Dies gilt allerdings nur für Aktionen, die am Frontend angestoßen werden. Bei dem aus dem ABAP angestoßen Suchen/Ersetzen hat die Selektion keinen Einfluß.

Tastaturkommando	Beschreibung
Strg+F Umsch+Strg+F Alt+Strg+F	Der Suchen und Ersetzen Dialog wird aufgerufen
Strg+G Umsch+Strg+G Alt+Ctrl+G	Der Suchen und Ersetzen Dialog springt zur nächsten Entsprechung.

Tastatur- und Maussteuerung im Editorfenster

Drag&Drop-Steuerung

Tastaturkommando	Beschreibung
Linke Maus auf selektiertem Text	Drag and Drop: Verschieben
Strg+Linke Maus auf selektiertem Text	Drag and Drop: Kopieren

Context-Menü

Tastaturkommando	Beschreibung
Umsch+F10	Kontextmenü
Rechte Maustaste	Kontextmenü

Textbereich ein- oder ausrücken

Tastaturkommando	Beschreibung
Tab	Einrücken der aktuellen Zeile / des selektierten Textbereiches
Umsch+Tab	Ausrücken der aktuellen Zeile / des selektierten Textbereiches

Text löschen

Tastaturkommando	Beschreibung
Rück	Rückwärts Löschen: falls Text selektiert ist, wird dieser gelöscht
Strg+Rück Umsch+Strg+Rück	wortweise rückwärts Löschen
Entf	Vorwärts Löschen: falls Text selektiert ist, wird dieser gelöscht
Strg+Entf Umsch+Strg+Entf	wortweise vorwärts Löschen

Methoden der Klasse CL_GUI_CFW**Methoden der Klasse CL_GUI_CFW**

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

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

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

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
control_list	Liste der Referenzvariablen zu aktiven Custom Controls. Liste ist vom Typ CNTO_CONTROL_LIST (in der Klasse CL_GUI_CFW definiert)

set_new_ok_code**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

Die Automation Queue wird durch den Aufruf der Methode [flush \[Seite 108\]](#) 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**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 \[Seite 113\]](#) relevant.

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

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

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
clsid	ID der Klasse
lifetime	<p>Parameter für das Lifetime Management. Folgende Werte sind möglich:</p> <p>my_control->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 118] aufgerufen.</p> <p>my_control->lifetime_dynpro: Das Control lebt, solange die Instanz des Dynpros existiert, d.h. sich noch im Dynprostapel befindet. Danach wird die Methode free [Seite 114] 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>my_control->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 ist), dann wird die Lebensdauer auf <code>my_control->lifetime_imode</code> gesetzt.</p>
shellstyle	<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>
parent	<p>Container, in dem das SAP Picture Control angezeigt werden kann (siehe SAP Container [Extern])</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 \[Seite 114\]](#) aufgerufen, bevor das Control am Frontend abgebaut wird.

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

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.

get_registered_events**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

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><code>my_control->state_dead</code>: Custom Control ist nicht mehr am Frontend aktiv</p> <p><code>my_control->state_alive</code>: Custom Control ist auf aktuellem Dynpro aktiv</p> <p><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)</p>

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_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**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

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**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

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**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

Methoden der Klasse CL_DRAGDROP

Die Klasse CL_DRAGDROP beinhaltet Methoden, über die das [Drag&Drop-Verhalten \[Extern\]](#) eines Custom Controls beschrieben wird.

constructor

constructor

Der Konstruktor erzeugt eine Instanz für die Beschreibung des Drag&Drop-Verhaltens eines Controls.

`CREATE OBJECT dragdrop.`

add

Diese Methode fügt eine weitere Beschreibung zu dem Drag&Drop-Verhalten hinzu. Es können beliebig viele Beschreibungen hinterlegt werden. Allerdings darf die gleiche Beschreibung nicht mehrmals hinzugefügt werden.

```
CALL METHOD dragdrop->add
    EXPORTING flavor          = flavor
             dragsrc         = dragsrc
             droptarget      = droptarget
             effect          = effect
             effect_in_ctrl  = effect_in_ctrl
    EXCEPTIONS allready_defined = 1
               obj_invalid     = 2.
```

Parameter	Bedeutung
flavor	Bezeichnung des neuen Flavors
dragsrc	'x': Beschreibung ist eine Drag-Quelle
droptarget	'x': Beschreibung ist ein Drop-Ziel
effect	Drop-Effekt der Beschreibung zwischen verschiedenen Custom Controls. Folgende Effekte werden unterstützt: dragdrop->copy: Darstellung der Maus beim Drag&Drop als Kopiervorgang dragdrop->move: Darstellung der Maus beim Drag&Drop als Verschiebevorgang dragdrop->none: Es ist kein Drag&Drop möglich
effect_in_ctrl	Drop-Effekt der Beschreibung im gleichen Custom Controls. Folgende Effekte werden unterstützt: dragdrop->copy: Darstellung der Maus beim Drag&Drop als Kopiervorgang dragdrop->move: Darstellung der Maus beim Drag&Drop als Verschiebevorgang dragdrop->none: Es ist kein Drag&Drop möglich dragdrop->use_default_effect: Es wird der gleiche Effekt benutzt, der durch den Parameter effect spezifiziert wurde

Ausnahmen	Bedeutung
allready_defined	Der angegebene Flavor wurde bereits definiert.
obj_invalid	Das Objekt wurde bereits mit der Methode destroy [Seite 134] zerstört

add

Wird bei der Definition des Flavors sowohl der Effekt `copy` als auch `move` benutzt, werden beim Drag&Drop-Vorgang bei normalem Drag die Flavors mit Effekt `Move` und beim Drag in Verbindung mit Drücken der Steuerungstaste die Flavors mit Effekt `Copy` verwendet.

clear

Der Inhalt der Instanz wird gelöscht. Nach Aufruf dieser Methode können keine Drag&Drop-Vorgänge mehr auf dem betroffenen Custom Control durchgeführt werden.

```
CALL METHOD dragdrop->clear  
EXCEPTIONS obj_invalid = 1.
```

Ausnahmen	Bedeutung
obj_invalid	Das Objekt wurde bereits mit der Methode destroy [Seite 134] zerstört

destroy

destroy

Der Inhalt der Instanz wird gelöscht. Weiterhin wird die Instanz zerstört. Nach Aufruf der Methode sind keine Drag&Drop-Vorgänge mehr auf dem betroffenen Custom Control möglich.

```
CALL METHOD dragdrop->destroy.
```

get

Diese Methode liefert die komplette Beschreibung zu einem Flavor zurück.

CALL METHOD dragdrop->get

```
EXPORTING flavor          = flavor
IMPORTING isdragsrc      = isdragsrc
          isdroptarget    = isdroptarget
          effect          = effect
          effect_in_ctrl  = effect_in_ctrl
EXCEPTIONS not_found    = 1
          obj_invalid    = 2.
```

Parameter	Bedeutung
flavor	Bezeichnung des Flavors
dragsrc	'x': Beschreibung ist eine Drag-Quelle
droptarget	'x': Beschreibung ist ein Drop-Ziel
effect	Drop-Effekt der Beschreibung zwischen verschiedenen Custom Controls. Folgende Effekte werden unterstützt: dragdrop->copy : Darstellung der Maus beim Drag&Drop als Kopiervorgang dragdrop->move : Darstellung der Maus beim Drag&Drop als Verschiebevorgang dragdrop->none : Es ist kein Drag&Drop möglich
effect_in_ctrl	Drop-Effekt der Beschreibung im gleichen Custom Controls. Folgende Effekte werden unterstützt: dragdrop->copy : Darstellung der Maus beim Drag&Drop als Kopiervorgang dragdrop->move : Darstellung der Maus beim Drag&Drop als Verschiebevorgang dragdrop->none : Es ist kein Drag&Drop möglich dragdrop->use_default_effect : Es wird der gleiche Effekt benutzt, der durch den Parameter effect spezifiziert wurde

Ausnahmen	Bedeutung
already_defined	Der angegebene Flavor wurde bereits definiert.



Wird bei der Definition des Flavors sowohl der Effekt **copy** als auch **move** benutzt, wird beim Drag&Drop-Vorgang bei normalem Drag die Flavors mit Effekt **move** und beim Drag in Verbindung mit Drücken der Steuerungstaste die Flavors mit Effekt **copy** verwendet.

get

get_handle

Die Methode liefert das Handle auf die Drag&Drop-Beschreibung. In den meisten Fällen ist ein Aufruf dieser Methode nicht nötig. Für tabellarische Massendatenschnittstellen (z.B. SAP Tree Control) muß dieses Handle aber vom Verwender des Controls in die Schnittstellentabelle kopiert werden.

```
CALL METHOD dragdrop->get_handle
    IMPORTING handle = handle
    EXCEPTIONS obj_invalid = 1.
```

Parameter	Bedeutung
handle	Handle auf die Drag&Drop-Beschreibung

Ausnahmen	Bedeutung
obj_invalid	Das Objekt wurde bereits mit der Methode destroy [Seite 134] zerstört

modify

modify

Mit dieser Methode können Sie einen bestehenden Flavor verändern.

```
CALL METHOD dragdrop->modify
  EXPORTING flavor          = flavor
           dragsrc         = dragsrc
           droptarget      = droptarget
           effect          = effect
           effect_in_ctrl  = effect_in_ctrl
  EXCEPTIONS not_found     = 1
            obj_invalid    = 2.
```

Parameter	Bedeutung
flavor	Bezeichnung des Flavors
dragsrc	'x': Beschreibung ist eine Drag-Quelle
droptarget	'x': Beschreibung ist ein Drop-Ziel
effect	Drop-Effekt der Beschreibung zwischen verschiedenen Custom Controls. Folgende Effekte werden unterstützt: dragdrop->copy : Darstellung der Maus beim Drag&Drop als Kopiervorgang dragdrop->move : Darstellung der Maus beim Drag&Drop als Verschiebevorgang dragdrop->none : Es ist kein Drag&Drop möglich
effect_in_ctrl	Drop-Effekt der Beschreibung im gleichen Custom Controls. Folgende Effekte werden unterstützt: dragdrop->copy : Darstellung der Maus beim Drag&Drop als Kopiervorgang dragdrop->move : Darstellung der Maus beim Drag&Drop als Verschiebevorgang dragdrop->none : Es ist kein Drag&Drop möglich dragdrop->use_default_effect : Es wird der gleiche Effekt benutzt, der durch den Parameter effect spezifiziert wurde

Ausnahmen	Bedeutung
not_found	Der angegebene Flavor existiert nicht
obj_invalid	Das Objekt wurde bereits mit der Methode destroy [Seite 134] zerstört



Wird bei der Definition des Flavors sowohl der Effekt **copy** als auch **move** benutzt, wird beim Drag&Drop-Vorgang bei normalem Drag die Flavors mit Effekt **move** und beim Drag in Verbindung mit Drücken der Steuerungstaste die Flavors mit Effekt **copy** verwendet.

remove

remove

Mit dieser Methode löschen Sie einen bestimmten Flavor.

```
CALL METHOD dragdrop->remove
  EXPORTING flavor = flavor
  EXCEPTIONS not_found = 1
             obj_invalid = 2.
```

Parameter	Bedeutung
flavor	Bezeichnung des Flavors

Ausnahmen	Bedeutung
not_found	Der angegebene Flavor existiert nicht
obj_invalid	Das Objekt wurde bereits mit der Methode destroy [Seite 134] zerstört

Methoden der Klasse CL_DRAGDROBJECT

Die Klasse CL_DRAGDROBJECT dient zur Beschreibung des Kontextes eines [Drag&Drop-Vorgangs \[Extern\]](#). Er beinhaltet in der Regel Informationen zu dem Quellobjekt, den Flavor des Drag&Drop-Vorgangs und Informationen zu Quelle und Ziel.

set_flavor**set_flavor**

Diese Methode verwenden Sie nur innerhalb der Ereignisbehandlung zu **ONGETFLAVOR**. Über den Parameter **newflavor** bestimmen Sie, welcher Flavor innerhalb des Drag&Drop-Vorgangs verwendet werden soll. Eine Liste der verfügbaren Flavors wird Ihnen als Ereignisparameter übergeben.

```
CALL METHOD dragdropobject->set_flavor
    EXPORTING newflavor = newflavor
    EXCEPTIONS illegal_state = 1
               illegal_flavor = 2.
```

Parameter	Bedeutung
newflavor	Bezeichnung des Flavors

Ausnahmen	Bedeutung
invalid_state	Die Methode wurde nicht innerhalb der Ereignisbehandlung ONGETFLAVOR aufgerufen.
obj_invalid	Es wurde ein Flavor verwendet, der nicht von der aktuellen Drag&Drop-Situation unterstützt wird.

abort

Der Drag&Drop-Vorgang wird sofort abgebrochen. Es werden keine weiteren Ereignisse ausgelöst.

```
CALL METHOD dragdropobject->abort.
```