

japi

Reference Manual

Merten Joost

Inhaltsverzeichnis

I	Reference	5
1	Components	7
	Button	7
	Borderpanel	11
	Canvas	16
	Checkbox	21
	CheckmenuItem	24
	Choice	25
	Dialog	28
	Focuslistener	34
	Frame	35
	Helpmenu	42
	Hscrollbar	44
	Graphicbutton	47
	Graphiclabel	50
	Image	53
	KeyListener	56
	Label	57
	Led	60
	List	63
	Menu	67
	MenuItem	69
	Meter	70
	Mouselistener	73
	Panel	74
	PopupMenu	79
	Printer	81
	Progressbar	84
	Radiobutton	87
	Sevensegment	90
	ScrollPane	93
	Textarea	96
	Textfield	100
	Vscrollbar	104
	Window	107

2 Functions	113
additem	113
add	115
alertbox	116
appendtext	117
beep	118
borderpanel	119
button	120
canvas	121
checkbox	122
checkmenuItem	123
choicebox2	124
choicebox3	125
choice	126
cliprect	127
componentlistener	128
connect	129
delete	130
deselect	131
dialog	132
disable	133
dispose	134
drawarc	135
drawcircle	136
drawimagesource	137
drawimage	138
drawline	139
drawoval	140
drawpixel	141
drawpolygon	142
drawpolyline	143
drawrect	144
drawroundrect	145
drawscaledimage	146
drawstring	147
enable	148
filedialog	149
fileselect	150
fillarc	151
fillcircle	152
filloval	153
fillpolygon	154
fillrect	155
fillroundrect	156
focuslistener	157
frame	158
getaction	159

getcolumns	160
getcurpos	161
getdanger	162
getfontascent	163
getfontheight	164
getheight	165
getimagesource	166
getimage	167
getinheight	168
getinsets	169
getinwidth	171
getitemcount	172
getitem	173
getkeychar	174
getkeycode	175
getlayoutid	176
getlength	177
getmousebutton	178
getmousex	179
getmousey	180
getparentid	181
getparent	182
getrows	183
getscaledimage	184
getscreenheight	185
getscreenwidth	186
getselect	187
getselend	188
getselstart	189
getseltext	190
getstate	191
getstringwidth	192
gettext	193
getvalue	194
getviewportheight	195
getviewportwidth	196
getwidth	197
getxpos	198
getypos	199
graphicbutton	200
graphiclabel	201
hasfocus	202
helpmenu	203
hide	204
hscrollbar	205
image	206
insert	207

inserttext	208
isparent	209
isresizable	210
isselect	211
isvisible	212
keylistener	213
label	214
led	215
line	216
list	218
loadimage	219
menubar	220
menuItem	221
menu	222
messagebox	223
meter	224
mouselistener	225
multiplemode	226
nextaction	227
pack	228
panel	229
popupmenu	230
printer	231
print	232
progressbar	233
quit	234
radiobutton	235
radiogroup	236
random	237
releaseall	238
release	239
removeall	240
removeitem	241
remove	242
replacetext	243
saveimage	244
scrollpane	245
selectall	246
select	247
selecttext	248
separator	249
setalign	250
setblockinc	251
setborderlayout	252
setborderpos	253
setcolorbg	254
setcolor	255

setcolumns	256
setcurpos	257
setcursor	258
setdanger	259
setdebug	260
setechochar	261
seteditable	262
setfixlayout	263
setflowfill	264
setflowlayout	265
setfocus	266
setfontname	267
setfontsize	268
setfontstyle	269
setfont	270
setgridlayout	272
sethgap	273
seticon	274
setimage	275
setinsets	276
setmax	277
setmin	278
setnamedcolorbg	279
setnamedcolor	280
setnolayout	281
start	282
setpos	283
setradiogroup	284
setresizable	285
setrows	286
setshortcut	287
setsize	288
setslidesize	289
setstate	290
settext	291
setunitinc	292
setvalue	293
setvgap	294
setxor	295
sevensegment	296
showpopup	297
show	298
sleep	299
start	300
sync	301
textarea	302
textfield	303

translate	304
vscrollbar	305
windowlistener	306
window	307

Teil I

Reference

Kapitel 1

Components

Button

- j_button** *integer function j_button (integer obj , character*(*) label)*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt den Button **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an den Button **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled den Button **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
der Button **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled den Button **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an den Button **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Button **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*

	Liefert die totale Höhe des aktuellen Fonts des Button obj in Bildschirmpixeln zurück.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Button obj zurück.
j_getlength	<i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes den Button zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Button obj in Bildschirmpixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Button .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Button obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Button obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Button obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt den Button obj nicht mehr auf dem Bildschirm an.
j_isplay	<i>integer function j_isplay (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an den Button obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an den Button obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i>

	Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Button .
j_release	<i>procedure j_release (integer obj)</i> Gibt den Button obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert den Button obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Button obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Button obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Button obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Button obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Button obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Button obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt den Button obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Button obj auf width and height .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Button obj auf str

j_show*procedure j_show (integer obj)*Zeigt den Button **obj** auf dem Bildschirm an.

Borderpanel

- j_borderpanel** *integer function j_borderpanel (integer obj , integer type)*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt das Borderpanel **obj** in den Container **cont** ein.
- j_borderpanel** *integer function j_borderpanel (integer obj , integer type)*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *integer function j_button (integer obj , character*(*) label)*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *integer function j_canvas (integer obj , integer width , integer height)*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *integer function j_checkbox (integer obj , character*(*) label)*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *integer function j_choice (integer obj)*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an das Borderpanel **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled das Borderpanel **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
das Borderpanel **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled das Borderpanel **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an das Borderpanel **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Borderpanel **obj** in Bildschirmpixeln zurück.

j_getfontheight	<i>integer function j_getfontheight (integer obj)</i> Liefert die totale Höhe des aktuellen Fonts des Borderpanel obj in Bildschirm-pixeln zurück.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Borderpanel obj zurück.
j_getinheight	<i>integer function j_getinheight (integer cont)</i> Liefert die Höhe des Borderpanel obj des inneren Bereichs zurück.
j_getinsets	<i>integer function j_getinsets (integer obj , integer side)</i> Liefert die angeforderte Breite des inneren Randes dem Borderpanel zurück.
j_getinwidth	<i>integer function j_getinwidth (integer cont)</i> Liefert die Breite des Borderpanel obj des inneren Bereichs zurück.
j_getlayoutid	<i>integer function j_getlayoutid (integer obj)</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Borderpanel obj in Bild-schirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Borderpanel obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Borderpanel obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Borderpanel obj zurück.
j_graphicbutton	<i>integer function j_graphicbutton (integer obj , character*(*) filename)</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>integer function j_graphiclabel (integer obj , character*(*) str)</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filena-me spezifiziert wird.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt das Borderpanel obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>integer function j_hscrollbar (integer obj)</i>

Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

- j_isparent** *integer function j_isparent (integer obj , integer cont)*
Liefert `.true.` zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst `.false.`
- j_isvisible** *integer function j_isvisible (integer obj)*
Liefert `.true.` zurück, falls **obj** sichtbar ist, sonst `.false.`
- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an das Borderpanel **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_label** *integer function j_label (integer obj , character*(*) label)*
Erzeugt ein neues Label Element mit dem Textinhalt **label**
- j_led** *integer function j_led (integer obj , integer style , integer color)*
Erzeugt eine LED der Farbe **color** und der Form **style**
- j_line** *integer function j_line (integer obj , integer orient , integer style , integer length)*
Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
- j_list** *integer function j_list (integer obj , integer rows)*
Erzeugt eine neue List und gibt deren Eventnummer zurück.
- j_meter** *integer function j_meter (integer obj , character*(*) title)*
Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung **title**.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an das Borderpanel **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_pack** *procedure j_pack (integer obj)*
Veranlaßt den Layoutmanager von **obj** die minimale Größe zu berechnen.
- j_panel** *integer function j_panel (integer obj)*
Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt des Borderpanel .
- j_progressbar** *integer function j_progressbar (integer obj , integer orient)*
Erzeugt eine Progress Bar mit der Ausrichtung **orient**.
- j_radiogroup** *integer function j_radiogroup (integer obj)*
Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.

j_releaseall	<i>procedure j_releaseall (integer obj)</i> Gibt alle Elemente aus dem Borderpanel obj frei.
j_release	<i>procedure j_release (integer obj)</i> Gibt das Borderpanel obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>integer function j_scrollpane (integer obj)</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (integer obj , integer align)</i> Setzt die Ausrichtung innerhalb des Borderpanel obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (integer obj)</i> Integriert einen Borderlayout Manager in das Borderpanel obj .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert das Borderpanel obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Borderpanel obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (integer obj)</i> Integriert einen Fixlayout Manager in das Borderpanel obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (integer obj , integer bool)</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Borderpanel obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (integer obj , integer align)</i> Integriert einen Flowlayout Manager in das Borderpanel obj mit der Ausrichtung align .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Borderpanel obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Borderpanel obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Borderpanel obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Borderpanel obj auf size .

j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Borderpanel obj auf style .
j_setgridlayout	<i>procedure j_setgridlayout (integer obj , integer row , integer col)</i> Integriert einen Gridlayout Manager in das Borderpanel obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (integer obj , integer hgap)</i> Setzt den horizontalen Abstand innerhalb des Borderpanel obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (integer obj)</i> Entfernt den aktuellen Layout Manager aus dem Borderpanel obj .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Borderpanel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Borderpanel obj auf width and height .
j_setvgap	<i>procedure j_setvgap (integer obj , integer vgap)</i> Setzt den vertikalen Abstand innerhalb des Borderpanel obj auf vgap Pixel.
j_sevensegment	<i>integer function j_sevensegment (integer obj , integer color)</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Borderpanel obj auf dem Bildschirm an.
j_textarea	<i>integer function j_textarea (integer obj , integer rows , integer columns)</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>integer function j_textfield (integer obj , integer columns)</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>integer function j_vscrollbar (integer obj)</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

Canvas

- j_canvas** *integer function j_canvas (integer obj , integer width , integer height)*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt den Canvas **obj** in den Container **cont** ein.
- j_cliprect** *procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)*
Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an den Canvas **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled den Canvas **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
der Canvas **obj** wird mit allen Recourcen gelöscht.
- j_drawarc** *procedure j_drawarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
Zeichnet einen Kreisbogen mit dem Mittelpunkt (**x**, **y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_drawcircle** *procedure j_drawcircle (integer obj , integer x , integer y , integer r)*
Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_drawimage** *procedure j_drawimage (integer obj , integer image , integer x , integer y)*
Kopiert das Image mit der Eventnummer **image** an die Position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Die Funktion stellt ein Bild an der Position (**x,y**) mit der Breite **w** und der Höhe **h** dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays **r,g,b** übergeben.
- j_drawline** *procedure j_drawline (integer obj , integer x1 , integer y1 , integer x2 , integer y2)*
Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (**x1,y1**) nach (**x2,y2**)

j_drawoval	<i>procedure j_drawoval (integer obj , integer x , integer y , integer rx , integer ry)</i> Zeichnet einen Oval mit dem Mittelpunkt (x,y) und den Hauptradien rx und ry .
j_drawpixel	<i>procedure j_drawpixel (integer obj , integer x , integer y)</i> Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (x,y) .
j_drawpolygon	<i>procedure j_drawpolygon (integer obj , integer len , array of integer x , array of integer y)</i> Zeichnet ein Polygon durch die Punkte, die durch die Arrays x und y gegeben sind.
j_drawpolyline	<i>procedure j_drawpolyline (integer obj , integer len , array of integer x , array of integer y)</i> Zeichnet eine Polylinie durch die Punkte, die durch die Arrays x und y gegeben sind.
j_drawrect	<i>procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)</i> Zeichnet ein Rechteck an die Position (x,y) mit der Breite width und der Höhe height .
j_drawroundrect	<i>procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)</i> Zeichnet ein Rechteck mit abgerundeten Ecken an die Position (x,y) mit der Breite width und der Höhe height . Die Radien der Ecken werden durch arcx und arcy bestimmt.
j_drawscaleddimage	<i>procedure j_drawscaleddimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)</i> Kopiert einen Ausschnitt des Bildes image , der durch die Parameter (sx, sy, sw, sh) bestimmt wird an die Position (tx, ty) und scaliert den Ausschnitt auf die Breite tw und die Höhe th
j_drawstring	<i>procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)</i> Zeichnet den String str an die durch die Parameter (x,y) festgelegte Koordinate.
j_enable	<i>procedure j_enable (integer obj)</i> enabled den Canvas obj .
j_fillarc	<i>procedure j_fillarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)</i> Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt (x,y) und den Hauptradien rx und ry . Die Parameter arc1 und arc2 bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
j_fillcircle	<i>procedure j_fillcircle (integer obj , integer x , integer y , integer r)</i> Zeichnet einen gefülltes Kreis mit dem Mittelpunkt (x,y) und dem Radius r .
j_filloval	<i>procedure j_filloval (integer obj , integer x , integer y , integer rx , integer ry)</i>

Zeichnet einen gefüllten Oval mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**.

- j_fillpolygon** *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
 Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *procedure j_fillrect (integer obj , integer x , integer y , integer width , integer height)*
 Zeichnet ein gefülltes Rechteck an die Position **(x,y)** mit der Breite **width** und der Höhe **height**.
- j_fillroundrect** *procedure j_fillroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
 Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
 Bindet an den Canvas **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
 Liefert die Oberlänge (Ascent) des aktuellen Fonts des Canvas **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
 Liefert die totale Höhe des aktuellen Fonts des Canvas **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
 Liefert die Höhe des Canvas **obj** zurück.
- j_getimage** *integer function j_getimage (integer obj)*
 Kopiert den Inhalt dem Canvas **obj** in ein Image und liefert dessen Eventnummer zurück.
- j_getimagesource** *integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
 Die Funktion liefert einen Ausschnitt vom Punkt **(x, y)** der Breite **w** und der Höhe **h** zurück. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays **r,g,b** abgelegt.
- j_getparentid** *integer function j_getparentid (integer obj)*
 Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
 Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird **-1** zurückgegeben.

j_getscaledimage	<i>integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)</i> Kopiert den Inhalt des Ausschnitts von Positio (x , y) it der Breite sw und der höhe sh in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von tw und einer Höhe von th scaliert.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Canvas obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Canvas obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Canvas obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Canvas obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt den Canvas obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert .true. zurück, falls obj sichtbar ist, sonst .false.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an den Canvas obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an den Canvas obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Canvas .
j_release	<i>procedure j_release (integer obj)</i> Gibt den Canvas obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert den Canvas obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Canvas obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Canvas obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Canvas obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Canvas obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Canvas obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Canvas obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt den Canvas obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Canvas obj auf width and height .
j_setxor	<i>procedure j_setxor (integer obj , integer bool)</i> Schaltet den Painting Mode auf XOR, falls bool = .true. ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
j_show	<i>procedure j_show (integer obj)</i> Zeigt den Canvas obj auf dem Bildschirm an.
j_translate	<i>procedure j_translate (integer obj , integer x , integer y)</i> Setzt den Ursprung des Koordinatensystems auf (x, y).

Checkbox

- j_checkbox** *integer function j_checkbox (integer obj , character*(*) label)*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Checkbox **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Checkbox **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Checkbox **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Checkbox **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Checkbox **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Checkbox **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Checkbox **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Checkbox **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Checkbox **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstate** *integer function j_getstate (integer obj)*
Liefert `.true.` , falls die Checkbox selektiert ist, sonst `.false.` .

j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts der Checkbox obj in Bildschirm-pixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt der Checkbox .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der Checkbox obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der Checkbox obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der Checkbox obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die Checkbox obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an die Checkbox obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an die Checkbox obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt der Checkbox .
j_release	<i>procedure j_release (integer obj)</i> Gibt die Checkbox obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert die Checkbox obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i>

	Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der Checkbox obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Checkbox obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Checkbox obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Checkbox obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Checkbox obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Checkbox obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Checkbox obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Checkbox obj auf width and height .
j_setstate	<i>procedure j_setstate (integer obj , integer bool)</i> Ist bool .true. so wird die Checkbox selectiert.
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt der Checkbox obj auf str
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Checkbox obj auf dem Bildschirm an.

Checkmenüitem

j_checkmenüitem	<i>integer function j_checkmenüitem (integer obj , character*(*) label)</i> Erzeugt ein Checkmenüitem mit der Beschriftung label und liefert dessen Eventnummer zurück.
j_disable	<i>procedure j_disable (integer obj)</i> disabled das Checkmenüitem obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>procedure j_dispose (integer obj)</i> das Checkmenüitem obj wird mit allen Recourcen gelöscht.
j_enable	<i>procedure j_enable (integer obj)</i> enabled das Checkmenüitem obj .
j_getlength	<i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes das Checkmenüitem zurück.
j_getstate	<i>integer function j_getstate (integer obj)</i> Liefert <i>.true.</i> , falls das Checkmenüitem selektiert ist, sonst <i>.false.</i> .
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Checkmenüitem .
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Checkmenüitem obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Checkmenüitem obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Checkmenüitem obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Checkmenüitem obj auf style .
j_setshortcut	<i>procedure j_setshortcut (integer obj , character chr)</i> Bestimmt das Zeichen chr als Shortcut für das Checkmenüitem .
j_setstate	<i>procedure j_setstate (integer obj , integer bool)</i> Ist bool <i>.true.</i> so wird das Checkmenüitem selectiert.
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Checkmenüitem obj auf str

Choice

- j_choice** *integer function j_choice (integer obj)*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_additem** *procedure j_additem (integer obj , character*(*) str)*
Hängt ein neues Item mit dem Inhalt **str** an die Auswahlliste an.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Choice **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Choice **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Choice **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Choice **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Choice **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Choice **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Choice **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Choice **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Choice **obj** zurück.
- j_getitemcount** *integer function j_getitemcount (integer obj)*
Liefert die Anzahl der Items der Choice **obj** zurück.
- j_getitem** *procedure j_getitem (integer obj , integer item , character*(*) str)*
liefert den Inhalt des Items mit dem Index **item** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.

j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselect	<i>integer function j_getselect (integer obj)</i> Liefert den Index des selectierten Items der Choice .
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts der Choice obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der Choice obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der Choice obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der Choice obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die Choice obj nicht mehr auf dem Bildschirm an.
j_insert	<i>integer function j_insert (integer obj , integer pos , character*(*) label)</i> fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert .true. zurück, falls obj sichtbar ist, sonst .false.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an die Choice obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an die Choice obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt der Choice .
j_release	<i>procedure j_release (integer obj)</i> Gibt die Choice obj aus dem umgebenden Container Element frei.
j_removeall	<i>integer function j_removeall (integer obj)</i> entfernt alle Items aus der Choice .

j_removeitem	<i>integer function j_removeitem (integer obj , character*(*) item)</i> entfernt den ersten Eintrag item aus der Choice .
j_remove	<i>integer function j_remove (integer obj , integer item)</i> entfernt das Item mit dem Index item aus der Choice .
j_select	<i>integer function j_select (integer obj , integer item)</i> selectiert das Item mit dem Index item .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert die Choice obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der Choice obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Choice obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Choice obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Choice obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Choice obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Choice obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Choice obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Choice obj auf width and height .
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Choice obj auf dem Bildschirm an.

Dialog

- j_dialog** *integer function j_dialog (integer obj , character*(*) label)*
Erzeugt ein neues Dialogwindow mit dem Titel **label** und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt den Dialog **obj** in den Container **cont** ein.
- j_borderpanel** *integer function j_borderpanel (integer obj , integer type)*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *integer function j_button (integer obj , character*(*) label)*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *integer function j_canvas (integer obj , integer width , integer height)*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *integer function j_checkbox (integer obj , character*(*) label)*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *integer function j_choice (integer obj)*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an den Dialog **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled den Dialog **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
der Dialog **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled den Dialog **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an den Dialog **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Dialog **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*

	Liefert die totale Höhe des aktuellen Fonts des Dialog obj in Bildschirmpixeln zurück.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Dialog obj zurück.
j_getinheight	<i>integer function j_getinheight (integer cont)</i> Liefert die Höhe des Dialog obj des inneren Bereichs zurück.
j_getinsets	<i>integer function j_getinsets (integer obj , integer side)</i> Liefert die angeforderte Breite des inneren Randes dem Dialog zurück.
j_getinwidth	<i>integer function j_getinwidth (integer cont)</i> Liefert die Breite des Dialog obj des inneren Bereichs zurück.
j_getlayoutid	<i>integer function j_getlayoutid (integer obj)</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getlength	<i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes den Dialog zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Dialog obj in Bildschirmpixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Dialog .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Dialog obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Dialog obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Dialog obj zurück.
j_graphicbutton	<i>integer function j_graphicbutton (integer obj , character*(*) filename)</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>integer function j_graphiclabel (integer obj , character*(*) str)</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.

j_hide	<i>procedure j_hide (integer obj)</i> Zeigt den Dialog obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>integer function j_hscrollbar (integer obj)</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <i>.true.</i> zurück, falls das cont die umgebende Komponente von obj ist, sonst <i>.false.</i>
j_isresizable	<i>integer function j_isresizable (integer obj)</i> Liefert <i>.false.</i> zurück, falls der Dialog eine fixe Größe besitzt, <i>.true.</i> sonst.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <i>.true.</i> zurück, falls obj sichtbar ist, sonst <i>.false.</i>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an den Dialog obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>integer function j_label (integer obj , character*(*) label)</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>integer function j_led (integer obj , integer style , integer color)</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>integer function j_line (integer obj , integer orient , integer style , integer length)</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>integer function j_list (integer obj , integer rows)</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>integer function j_meter (integer obj , character*(*) title)</i> Erzeugt ein analoges Anzeigeelement mit der Beschriftung title .
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an den Dialog obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (integer obj)</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>integer function j_panel (integer obj)</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Dialog .

j_progressbar	<i>integer function j_progressbar (integer obj , integer orient)</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>integer function j_radiogroup (integer obj)</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (integer obj)</i> Gibt alle Elemente aus dem Dialog obj frei.
j_release	<i>procedure j_release (integer obj)</i> Gibt den Dialog obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>integer function j_scrollpane (integer obj)</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (integer obj , integer align)</i> Setzt die Ausrichtung innerhalb des Dialog obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (integer obj)</i> Integriert einen BorderLayout Manager in den Dialog obj .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert den Dialog obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Dialog obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (integer obj)</i> Integriert einen Fixlayout Manager in den Dialog obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (integer obj , integer bool)</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Dialog obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (integer obj , integer align)</i> Integriert einen Flowlayout Manager in den Dialog obj mit der Ausrichtung align .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Dialog obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Dialog obj auf name .

j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Dialog obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Dialog obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Dialog obj auf style .
j_setgridlayout	<i>procedure j_setgridlayout (integer obj , integer row , integer col)</i> Integriert einen Gridlayout Manager in den Dialog obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (integer obj , integer hgap)</i> Setzt den horizontalen Abstand innerhalb des Dialog obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (integer obj)</i> Entfernt den aktuellen Layout Manager aus dem Dialog obj .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt den Dialog obj zur angegebenen Koordinate (xpos,ypos) an.
j_setresizable	<i>procedure j_setresizable (integer obj , integer resizable)</i> Ist resizable .false. , kann die Größe des Dialog nicht vom Benutzer verändert werden.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Dialog obj auf width and height .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Dialog obj auf str
j_setvgap	<i>procedure j_setvgap (integer obj , integer vgap)</i> Setzt den vertikalen Abstand innerhalb des Dialog obj auf vgap Pixel.
j_sevensegment	<i>integer function j_sevensegment (integer obj , integer color)</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (integer obj)</i> Zeigt den Dialog obj auf dem Bildschirm an.
j_textarea	<i>integer function j_textarea (integer obj , integer rows , integer columns)</i>

Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.

j_textfield

integer function j_textfield (integer obj , integer columns)

Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.

j_vscrollbar

integer function j_vscrollbar (integer obj)

Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

j_windowlistener

integer function j_windowlistener (integer window , integer kind)

Bindet an den Dialog **window** einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.

Focuslistener

- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an den Focuslistener **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_dispose** *procedure j_dispose (integer obj)*
der Focuslistener **obj** wird mit allen Recourcen gelöscht.
- j_hasfocus** *integer function j_hasfocus (integer obj)*
Liefert *.true.* zurück, falls der Focuslistener den Focus besitzt, sonst *.false.* .

Frame

- j_frame** *integer function j_frame (character*(*) label)*
Erzeugt einen Frame mit der Titelzeile **label**.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt den Frame **obj** in den Container **cont** ein.
- j_alertbox** *procedure j_alertbox (integer obj , character*(*) title , character*(*) text , character*(*) button)*
Öffnet eine Alertbox mit der Überschrift **title**, Textinhalt **text** und einem Button.
- j_borderpanel** *integer function j_borderpanel (integer obj , integer type)*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *integer function j_button (integer obj , character*(*) label)*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *integer function j_canvas (integer obj , integer width , integer height)*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *integer function j_checkbox (integer obj , character*(*) label)*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choicebox2** *procedure j_choicebox2 (integer obj , character*(*) title , character*(*) text , character*(*) button1 , character*(*) button2)*
Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und zwei Buttons.
- j_choicebox3** *procedure j_choicebox3 (integer obj , character*(*) title , character*(*) text , character*(*) button1 , character*(*) button2 , character*(*) button3)*
Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und drei Buttons.
- j_choice** *integer function j_choice (integer obj)*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an den Frame **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_dialog** *integer function j_dialog (integer obj , character*(*) label)*
Erzeugt ein neues Dialogwindow mit dem Titel **label** und liefert dessen Eventnummer zurück.

j_disable	<i>procedure j_disable (integer obj)</i> disabled den Frame obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>procedure j_dispose (integer obj)</i> der Frame obj wird mit allen Recourcen gelöscht.
j_enable	<i>procedure j_enable (integer obj)</i> enabled den Frame obj .
j_filedialog	<i>procedure j_filedialog (integer frame , character*(*) title , character*(*) directory , character*(*) filename)</i> Öffnet eine Filedialogbox in dem angegebenen directory mit dem Title title und liefert den ausgewählten Dateiname in filename zurück.
j_fileselect	<i>procedure j_fileselect (integer frame , character*(*) title , character*(*) filter , character*(*) filename)</i> Öffnet eine Fileslectorbox mit dem voreingestellten filename und dem Title title und liefert den ausgewählten Dateiname in filename zurück.
j_focuslistener	<i>integer function j_focuslistener (integer obj)</i> Bindet an den Frame obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
j_getfontascent	<i>integer function j_getfontascent (integer obj)</i> Liefert die Oberlänge (Ascent) des aktuellen Fonts des Frame obj in Bildschirmpixeln zurück.
j_getfontheight	<i>integer function j_getfontheight (integer obj)</i> Liefert die totale Höhe des aktuellen Fonts des Frame obj in Bildschirmpixeln zurück.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Frame obj zurück.
j_getinheight	<i>integer function j_getinheight (integer cont)</i> Liefert die Höhe des Frame obj des inneren Bereichs zurück.
j_getinsets	<i>integer function j_getinsets (integer obj , integer side)</i> Liefert die angeforderte Breite des inneren Randes dem Frame zurück.
j_getinwidth	<i>integer function j_getinwidth (integer cont)</i> Liefert die Breite des Frame obj des inneren Bereichs zurück.
j_getlayoutid	<i>integer function j_getlayoutid (integer obj)</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getlength	<i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes den Frame zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.

j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Frame obj in Bildschirmpixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Frame .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Frame obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Frame obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Frame obj zurück.
j_graphicbutton	<i>integer function j_graphicbutton (integer obj , character*(*) filename)</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>integer function j_graphiclabel (integer obj , character*(*) str)</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt den Frame obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>integer function j_hscrollbar (integer obj)</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_isresizable	<i>integer function j_isresizable (integer obj)</i> Liefert .false. zurueck, falls der Frame eine fixe Größe besitzt, .true. sonst.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert .true. zurück, falls obj sichtbar ist, sonst .false.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an den Frame obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>integer function j_label (integer obj , character*(*) label)</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>integer function j_led (integer obj , integer style , integer color)</i> Erzeugt eine LED der Farbe color und der Form style

j_line	<i>integer function j_line (integer obj , integer orient , integer style , integer length)</i> Erzeugt ein neues Linielement und liefert dessen Eventnummer zurück
j_list	<i>integer function j_list (integer obj , integer rows)</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_menubar	<i>integer function j_menubar (integer obj)</i> Erzeugt eine neue Menubar in obj und liefert dessen Eventnummer zurück.
j_messagebox	<i>procedure j_messagebox (integer obj , character*(*) title , character*(*) text)</i> Öffnet eine MessageBox mit der Überschrift title und dem Textinhalt text
j_meter	<i>integer function j_meter (integer obj , character*(*) title)</i> Erzeugt ein analoges Anzeigeelement mit der Beschriftung title .
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an den Frame obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (integer obj)</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>integer function j_panel (integer obj)</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popuptaste mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_printer	<i>integer function j_printer (integer frame)</i> Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Frame .
j_progressbar	<i>integer function j_progressbar (integer obj , integer orient)</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>integer function j_radiogroup (integer obj)</i> Erzeugt eine neue Radiogruppe und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (integer obj)</i> Gibt alle Elemente aus dem Frame obj frei.
j_release	<i>procedure j_release (integer obj)</i> Gibt den Frame obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>integer function j_scrollpane (integer obj)</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (integer obj , integer align)</i>

	Setzt die Ausrichtung innerhalb des Frame obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (integer obj)</i> Integriert einen BorderLayout Manager in den Frame obj .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert den Frame obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Frame obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (integer obj)</i> Integriert einen Fixlayout Manager in den Frame obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (integer obj , integer bool)</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Frame obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (integer obj , integer align)</i> Integriert einen Flowlayout Manager in den Frame obj mit der Ausrichtung align .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Frame obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Frame obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Frame obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Frame obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Frame obj auf style .
j_setgridlayout	<i>procedure j_setgridlayout (integer obj , integer row , integer col)</i> Integriert einen Gridlayout Manager in den Frame obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (integer obj , integer hgap)</i> Setzt den horizontalen Abstand innerhalb des Frame obj auf hgap Pixel.

j_seticon	<i>procedure j_seticon (integer frame , integer icon)</i> Ordnet dem frame das Icon icon zu. Eine iconisierte Darstellung wird nicht von allen Plattformen unterstützt.
j_setinsets	<i>procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (integer obj)</i> Entfernt den aktuellen Layout Manager aus dem Frame obj .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt den Frame obj zur angegebenen Koordinate (xpos,ypos) an.
j_setresizable	<i>procedure j_setresizable (integer obj , integer resizable)</i> Ist resizable .false. , kann die Größe des Frame nicht vom Benutzer verändert werden.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Frame obj auf width and height .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Frame obj auf str
j_setvgap	<i>procedure j_setvgap (integer obj , integer vgap)</i> Setzt den vertikalen Abstand innerhalb des Frame obj auf vgap Pixel.
j_sevensegment	<i>integer function j_sevensegment (integer obj , integer color)</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (integer obj)</i> Zeigt den Frame obj auf dem Bildschirm an.
j_textarea	<i>integer function j_textarea (integer obj , integer rows , integer columns)</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>integer function j_textfield (integer obj , integer columns)</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>integer function j_vscrollbar (integer obj)</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>integer function j_windowlistener (integer window , integer kind)</i> Bindet an den Frame window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.

j_window*integer function j_window (integer obj)*

Erzeugt ein neues Window und liefert dessen Eventnummer zurück.

Helpmenu

j_helpmenu	<p><i>integer function j_helpmenu (integer obj , character*(*) label)</i> Erzeugt ein Hilfe-Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_checkmenuitem	<p><i>integer function j_checkmenuitem (integer obj , character*(*) label)</i> Erzeugt ein Checkmenuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_disable	<p><i>procedure j_disable (integer obj)</i> disabled das Helpmenu obj, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
j_dispose	<p><i>procedure j_dispose (integer obj)</i> das Helpmenu obj wird mit allen Recourcen gelöscht.</p>
j_enable	<p><i>procedure j_enable (integer obj)</i> enabled das Helpmenu obj.</p>
j_getlength	<p><i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes das Helpmenu zurück.</p>
j_gettext	<p><i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Helpmenu .</p>
j_menuitem	<p><i>integer function j_menuitem (integer obj , character*(*) label)</i> Erzeugt ein Menuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_seperator	<p><i>procedure j_seperator (integer obj)</i> Erzeugt eine Trennlinie in das Helpmenu .</p>
j_setfontname	<p><i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Helpmenu obj auf name.</p>
j_setfont	<p><i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Helpmenu obj entsprechend der Parameterliste.</p>
j_setfontsize	<p><i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Helpmenu obj auf size.</p>
j_setfontstyle	<p><i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Helpmenu obj auf style.</p>
j_setshortcut	<p><i>procedure j_setshortcut (integer obj , character chr)</i> Bestimmt das Zeichen chr als Shortcut für das Helpmenu .</p>
j_settext	<p><i>procedure j_settext (integer obj , character*(*) str)</i></p>

Ändert den Textinhalt des Helpmenu **obj** auf **str**

Hscrollbar

- j_hscrollbar** *integer function j_hscrollbar (integer obj)*
Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Hscrollbar **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Hscrollbar **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Hscrollbar **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Hscrollbar **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Hscrollbar **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Hscrollbar **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Hscrollbar **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Hscrollbar **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Hscrollbar **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*
Liefert die Länge des **str** des aktuellen Fonts der Hscrollbar **obj** in Bildschirmpixeln zurück.

j_getvalue	<i>integer function j_getvalue (integer obj)</i> Liefert den aktuellen Wert der Hscrollbar .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der Hscrollbar obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der Hscrollbar obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der Hscrollbar obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die Hscrollbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <i>.true.</i> zurück, falls das cont die umgebende Komponente von obj ist, sonst <i>.false.</i>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <i>.true.</i> zurück, falls obj sichtbar ist, sonst <i>.false.</i>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an die Hscrollbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an die Hscrollbar obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt der Hscrollbar .
j_release	<i>procedure j_release (integer obj)</i> Gibt die Hscrollbar obj aus dem umgebenden Container Element frei.
j_setblockinc	<i>integer function j_setblockinc (integer obj , integer val)</i> Stellt die Blockänderung von der Hscrollbar auf den Wert von val ein.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert die Hscrollbar obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der Hscrollbar obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Hscrollbar obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Hscrollbar obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Hscrollbar obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Hscrollbar obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Hscrollbar obj auf style .
j_setmax	<i>integer function j_setmax (integer obj , integer val)</i> Stellt die obere Grenze der Hscrollbar auf den Wert von val ein.
j_setmin	<i>integer function j_setmin (integer obj , integer val)</i> Stellt die untere Grenze der Hscrollbar auf den Wert von val ein.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Hscrollbar obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Hscrollbar obj auf width and height .
j_setslidesize	<i>integer function j_setslidesize (integer obj , integer val)</i> Setzt die Größe des Schiebers auf den Wert von val .
j_setunitinc	<i>integer function j_setunitinc (integer obj , integer val)</i> Stellt die minimale Wertänderung von die Hscrollbar auf den Wert von val ein.
j_setvalue	<i>procedure j_setvalue (integer obj , integer val)</i> Stellt den aktuellen Wert der Hscrollbar obj auf val .
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Hscrollbar obj auf dem Bildschirm an.

Graphicbutton

- j_graphicbutton** *integer function j_graphicbutton (integer obj , character*(*) filename)*
Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch **filename** spezifiziert wird.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt den Graphicbutton **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an den Graphicbutton **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled den Graphicbutton **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
der Graphicbutton **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled den Graphicbutton **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an den Graphicbutton **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Graphicbutton **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts des Graphicbutton **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe des Graphicbutton **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*

Liefert die Länge des **str** des aktuellen Fonts des Graphicbutton **obj** in Bildschirmpixeln zurück.

- j_getwidth** *integer function j_getwidth (integer obj)*
Liefert die Breite des Graphicbutton **obj** zurück.
- j_getxpos** *integer function j_getxpos (integer obj)*
Liefert die aktuelle horizontale Position des Graphicbutton **obj** zurück.
- j_getypos** *integer function j_getypos (integer obj)*
Liefert die aktuelle vertikale Position des Graphicbutton **obj** zurück.
- j_hide** *procedure j_hide (integer obj)*
Zeigt den Graphicbutton **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *integer function j_isparent (integer obj , integer cont)*
Liefert **.true.** zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst **.false.**
- j_isvisible** *integer function j_isvisible (integer obj)*
Liefert **.true.** zurück, falls **obj** sichtbar ist, sonst **.false.**
- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an den Graphicbutton **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an den Graphicbutton **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt des Graphicbutton .
- j_release** *procedure j_release (integer obj)*
Gibt den Graphicbutton **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (integer obj , integer pos)*
Positioniert den Graphicbutton **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (integer obj , integer r , integer g , , integer b)*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (integer obj , integer r , integer g , , integer b)*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcursor** *integer function j_setcursor (integer obj , integer cursor)*
Setzt den Cursor des Graphicbutton **obj** auf **cursor**.

j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Graphicbutton obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Graphicbutton obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Graphicbutton obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Graphicbutton obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Graphicbutton obj auf style .
j_setimage	<i>procedure j_setimage (integer obj , integer image)</i> Ordnet dem obj das Bild image zu.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt den Graphicbutton obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Graphicbutton obj auf width and height .
j_show	<i>procedure j_show (integer obj)</i> Zeigt den Graphicbutton obj auf dem Bildschirm an.

Graphiclabel

- j_graphiclabel** *integer function j_graphiclabel (integer obj , character*(*) str)*
Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch **filena-**
me spezifiziert wird.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt das Graphiclabel **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an das Graphiclabel **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled das Graphiclabel **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
das Graphiclabel **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled das Graphiclabel **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an das Graphiclabel **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Graphiclabel **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts des Graphiclabel **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe des Graphiclabel **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*

Liefert die Länge des **str** des aktuellen Fonts des Graphiclabel **obj** in Bildschirmpixeln zurück.

- j_getwidth** *integer function j_getwidth (integer obj)*
Liefert die Breite des Graphiclabel **obj** zurück.
- j_getxpos** *integer function j_getxpos (integer obj)*
Liefert die aktuelle horizontale Position des Graphiclabel **obj** zurück.
- j_getypos** *integer function j_getypos (integer obj)*
Liefert die aktuelle vertikale Position des Graphiclabel **obj** zurück.
- j_hide** *procedure j_hide (integer obj)*
Zeigt das Graphiclabel **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *integer function j_isparent (integer obj , integer cont)*
Liefert **.true.** zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst **.false.**
- j_isvisible** *integer function j_isvisible (integer obj)*
Liefert **.true.** zurück, falls **obj** sichtbar ist, sonst **.false.**
- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an das Graphiclabel **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an das Graphiclabel **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt des Graphiclabel .
- j_release** *procedure j_release (integer obj)*
Gibt das Graphiclabel **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (integer obj , integer pos)*
Positioniert das Graphiclabel **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (integer obj , integer r , integer g , , integer b)*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (integer obj , integer r , integer g , , integer b)*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcursor** *integer function j_setcursor (integer obj , integer cursor)*
Setzt den Cursor des Graphiclabel **obj** auf **cursor**.

j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Graphiclabel obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Graphiclabel obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Graphiclabel obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Graphiclabel obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Graphiclabel obj auf style .
j_setimage	<i>procedure j_setimage (integer obj , integer image)</i> Ordnet dem obj das Bild image zu.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Graphiclabel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Graphiclabel obj auf width and height .
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Graphiclabel obj auf dem Bildschirm an.

Image

- j_image** *integer function j_image (integer width , integer height)*
Erzeugt einen neues (Memory) Image mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_cliprect** *procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)*
Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).
- j_dispose** *procedure j_dispose (integer obj)*
das Image **obj** wird mit allen Recourcen gelöscht.
- j_drawarc** *procedure j_drawarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
Zeichnet einen Kreisbogen mit dem Mittelpunkt (**x**, **y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_drawcircle** *procedure j_drawcircle (integer obj , integer x , integer y , integer r)*
Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_drawimage** *procedure j_drawimage (integer obj , integer image , integer x , integer y)*
Kopiert das Image mit der Eventnummer **image** an die Position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Die Funktion stellt ein Bild an der Position (**x,y**) mit der Breite **w** und der Höhe **h** dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays r,g,b übergeben.
- j_drawline** *procedure j_drawline (integer obj , integer x1 , integer y1 , integer x2 , integer y2)*
Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (**x1,y1**) nach (**x2,y2**)
- j_drawoval** *procedure j_drawoval (integer obj , integer x , integer y , integer rx , integer ry)*
Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_drawpixel** *procedure j_drawpixel (integer obj , integer x , integer y)*
Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (**x,y**).
- j_drawpolygon** *procedure j_drawpolygon (integer obj , integer len , array of integer x , array of integer y)*

- Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawpolyline** *procedure j_drawpolyline (integer obj , integer len , array of integer x , array of integer y)*
Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawrect** *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
Zeichnet ein Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
Zeichnet ein Rechteck mit abgerundeten Ecken an die Position (**x,y**) mit der Breite **width** und der Höhe **height**. Die Radien der Ecken werden durch **arcx** und **arcy** bestimmt.
- j_drawscaledimage** *procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)*
Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter (**sx**, **sy**, **sw**, **sh**) bestimmt wird an die Position (**tx**, **ty**) und scaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**
- j_drawstring** *procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)*
Zeichnet den String **str** an die durch die Parameter (**x,y**) festgelegte Koordinate.
- j_fillarc** *procedure j_fillarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_fillcircle** *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_filloval** *procedure j_filloval (integer obj , integer x , integer y , integer rx , integer ry)*
Zeichnet einen gefüllten Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_fillpolygon** *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *procedure j_fillrect (integer obj , integer x , integer y , integer width , integer height)*
Zeichnet ein gefülltes Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.

j_fillroundrect	<i>procedure j_fillroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)</i> Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position (x,y). Das Rechteck hat eine Breite von width und eine Höhe von height Pixeln. Die Form der runden Ecken wird durch die Parameter arcx und arcy bestimmt.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Image obj zurück.
j_getimage	<i>integer function j_getimage (integer obj)</i> Kopiert den Inhalt dem Image obj in ein Image und liefert dessen Eventnummer zurück.
j_getimagesource	<i>integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)</i> Die Funktion liefert einen Ausschnitt vom Punkt (x, y) der Breite w und der Höhe h zurück. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays r,g,b abgelegt.
j_getscaledimage	<i>integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)</i> Kopiert den Inhalt des Ausschnitts von Positio (x, y) it der Breite sw und der höhe sh in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von tw und einer Höhe von th skaliert.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Image obj zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Image .
j_setxor	<i>procedure j_setxor (integer obj , integer bool)</i> Schaltet den Painting Mode auf XOR, falls bool = .true. ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
j_translate	<i>procedure j_translate (integer obj , integer x , integer y)</i> Setzt den Ursprung des Koordinatensystems auf (x, y).

KeyListenerer

- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an den Keylistener **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_dispose** *procedure j_dispose (integer obj)*
der Keylistener **obj** wird mit allen Recourcen gelöscht.
- j_getkeychar** *integer function j_getkeychar (integer obj)*
Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.
- j_getkeycode** *integer function j_getkeycode (integer obj)*
Liefert den Keycode, der zuletzt gedrückten Taste zurück.

Label

- j_label** *integer function j_label (integer obj , character*(*) label)*
Erzeugt ein neues Label Element mit dem Textinhalt **label**
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt das Label **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an das Label **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled das Label **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
das Label **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled das Label **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an das Label **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Label **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts des Label **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe des Label **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*
Liefert die Länge des **str** des aktuellen Fonts des Label **obj** in Bildschirmpixeln zurück.

j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Label .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Label obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Label obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Label obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt das Label obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <i>.true.</i> zurück, falls das cont die umgebende Komponente von obj ist, sonst <i>.false.</i>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <i>.true.</i> zurück, falls obj sichtbar ist, sonst <i>.false.</i>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an das Label obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an das Label obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Label .
j_release	<i>procedure j_release (integer obj)</i> Gibt das Label obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert das Label obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Label obj auf cursor .

j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Label obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Label obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Label obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Label obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Label obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Label obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Label obj auf width and height .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Label obj auf str
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Label obj auf dem Bildschirm an.

Led

- j_led** *integer function j_led (integer obj , integer style , integer color)*
Erzeugt eine LED der Farbe **color** und der Form **style**
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die LED **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die LED **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die LED **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die LED **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die LED **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die LED **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der LED **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der LED **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der LED **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstate** *integer function j_getstate (integer obj)*
Liefert `.true.` , falls die LED selektiert ist, sonst `.false.` .
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*

	Liefert die Länge des str des aktuellen Fonts der LED obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der LED obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der LED obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der LED obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die LED obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an die LED obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an die LED obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt der LED .
j_release	<i>procedure j_release (integer obj)</i> Gibt die LED obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert die LED obj an die angegebene Position pos . Im umgebenden Container muß ein BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der LED obj auf cursor .

j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der LED obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der LED obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der LED obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der LED obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der LED obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die LED obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der LED obj auf width and height .
j_setstate	<i>procedure j_setstate (integer obj , integer bool)</i> Ist bool .true . so wird die LED selectiert.
j_show	<i>procedure j_show (integer obj)</i> Zeigt die LED obj auf dem Bildschirm an.

List

- j_list** *integer function j_list (integer obj , integer rows)*
Erzeugt eine neue List und gibt deren Eventnummer zurück.
- j_additem** *procedure j_additem (integer obj , character*(*) str)*
Hängt ein neues Item mit dem Inhalt **str** an die Auswahlliste an.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die List **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die List **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_deselect** *integer function j_deselect (integer obj , integer item)*
Deselectiert das Item mit dem Index **item**.
- j_disable** *procedure j_disable (integer obj)*
disabled die List **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die List **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die List **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die List **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der List **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der List **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der List **obj** zurück.
- j_getitemcount** *integer function j_getitemcount (integer obj)*
Liefert die Anzahl der Items der List **obj** zurück.
- j_getitem** *procedure j_getitem (integer obj , integer item , character*(*) str)*
liefert den Inhalt des Items mit dem Index **item** zurück.

j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselect	<i>integer function j_getselect (integer obj)</i> Liefert den Index des selectierten Items der List .
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts der List obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der List obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der List obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der List obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die List obj nicht mehr auf dem Bildschirm an.
j_insert	<i>integer function j_insert (integer obj , integer pos , character*(*) label)</i> fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_iselect	<i>integer function j_iselect (integer obj , integer item)</i> Liefert .true. zurück, falls das Item mit dem Index item selectiert ist, sonst .false.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert .true. zurück, falls obj sichtbar ist, sonst .false.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an die List obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an die List obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_multiplemode	<i>integer function j_multiplemode (integer obj , integer bool)</i> die List erlaubt mehrfache Selectierung, falls bool .true. ist.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i>

	Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt der List .
j_release	<i>procedure j_release (integer obj)</i> Gibt die List obj aus dem umgebenden Container Element frei.
j_removeall	<i>integer function j_removeall (integer obj)</i> entfernt alle Items aus der List .
j_removeitem	<i>integer function j_removeitem (integer obj , character*(*) item)</i> entfernt den ersten Eintrag item aus der List .
j_remove	<i>integer function j_remove (integer obj , integer item)</i> entfernt das Item mit dem Index item aus der List .
j_select	<i>integer function j_select (integer obj , integer item)</i> selectiert das Item mit dem Index item .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert die List obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der List obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der List obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der List obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der List obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der List obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der List obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .

- j_setnamedcolor** *procedure j_setnamedcolor (integer obj , integer color)*
Setzt die Vordergrundfarbe auf eine vordefinierte Farbe **color**.
- j_setpos** *procedure j_setpos (integer obj , integer xpos , integer ypos)*
Verschiebt die List **obj** zur angegebenen Koordinate (**xpos,ypos**) an.
- j_setsize** *procedure j_setsize (integer obj , integer width , integer height)*
Setzt Breite und Höhe der List **obj** auf **width** and **height**.
- j_show** *procedure j_show (integer obj)*
Zeigt die List **obj** auf dem Bildschirm an.

Menu

j_menu	<p><i>integer function j_menu (integer obj , character*(*) str)</i> Erzeugt ein Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_checkmenuitem	<p><i>integer function j_checkmenuitem (integer obj , character*(*) label)</i> Erzeugt ein Checkmenuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_disable	<p><i>procedure j_disable (integer obj)</i> disabled das Menu obj, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
j_dispose	<p><i>procedure j_dispose (integer obj)</i> das Menu obj wird mit allen Recourcen gelöscht.</p>
j_enable	<p><i>procedure j_enable (integer obj)</i> enabled das Menu obj.</p>
j_getlength	<p><i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes das Menu zurück.</p>
j_gettext	<p><i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Menu .</p>
j_helpmenu	<p><i>integer function j_helpmenu (integer obj , character*(*) label)</i> Erzeugt ein Hilfe-Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_menuitem	<p><i>integer function j_menuitem (integer obj , character*(*) label)</i> Erzeugt ein MenuItem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_menu	<p><i>integer function j_menu (integer obj , character*(*) str)</i> Erzeugt ein Menueintrag mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_seperator	<p><i>procedure j_seperator (integer obj)</i> Erzeugt eine Trennlinie in das Menu .</p>
j_setfontname	<p><i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Menu obj auf name.</p>
j_setfont	<p><i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Menu obj entsprechend der Parameterliste.</p>
j_setfontsize	<p><i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Menu obj auf size.</p>

j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Menu obj auf style .
j_setshortcut	<i>procedure j_setshortcut (integer obj , character chr)</i> Bestimmt das Zeichen chr als Shortcut für das Menu .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Menu obj auf str

Menuitem

j_menuitem	<i>integer function j_menuitem (integer obj , character*(*) label)</i> Erzeugt ein Menuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.
j_disable	<i>procedure j_disable (integer obj)</i> disabled das Menuitem obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>procedure j_dispose (integer obj)</i> das Menuitem obj wird mit allen Recourcen gelöscht.
j_enable	<i>procedure j_enable (integer obj)</i> enabled das Menuitem obj .
j_getlength	<i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes das Menuitem zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Menuitem .
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Menuitem obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Menuitem obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Menuitem obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Menuitem obj auf style .
j_setshortcut	<i>procedure j_setshortcut (integer obj , character chr)</i> Bestimmt das Zeichen chr als Shortcut für das Menuitem .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Menuitem obj auf str

Meter

- j_meter** *integer function j_meter (integer obj , character*(*) title)*
Erzeugt ein analoges Anzeigeeinstrument mit der Beschriftung **title**.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt das Anzeigeeinstrument **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an das Anzeigeeinstrument **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled das Anzeigeeinstrument **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
das Anzeigeeinstrument **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled das Anzeigeeinstrument **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an das Anzeigeeinstrument **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getdanger** *procedure j_getdanger (integer obj)*
Liefert den Warnlevel des Anzeigeeinstrument **obj** zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Anzeigeeinstrument **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts des Anzeigeeinstrument **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe des Anzeigeeinstrument **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.

j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Anzeigeinstrument obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Anzeigeinstrument obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Anzeigeinstrument obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Anzeigeinstrument obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt das Anzeigeinstrument obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert .true. zurück, falls obj sichtbar ist, sonst .false.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an das Anzeigeinstrument obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an das Anzeigeinstrument obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popuptmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Anzeigeinstrument .
j_release	<i>procedure j_release (integer obj)</i> Gibt das Anzeigeinstrument obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert das Anzeigeinstrument obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i>

	Setzt den Cursor des Anzeigeeinstrument obj auf cursor .
j_setdanger	<i>procedure j_setdanger (integer obj , integer val)</i> Stellt den Warnlevel dem Anzeigeeinstrument obj auf val .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Anzeigeeinstrument obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Anzeigeeinstrument obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Anzeigeeinstrument obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Anzeigeeinstrument obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Anzeigeeinstrument obj auf style .
j_setmax	<i>integer function j_setmax (integer obj , integer val)</i> Stellt die obere Grenze dem Anzeigeeinstrument auf den Wert von val ein.
j_setmin	<i>integer function j_setmin (integer obj , integer val)</i> Stellt die untere Grenze dem Anzeigeeinstrument auf den Wert von val ein.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Anzeigeeinstrument obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Anzeigeeinstrument obj auf width and height .
j_setvalue	<i>procedure j_setvalue (integer obj , integer val)</i> Stellt den aktuellen Wert dem Anzeigeeinstrument obj auf val .
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Anzeigeeinstrument obj auf dem Bildschirm an.

Mouselistener

- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an den Mouselistener **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_dispose** *procedure j_dispose (integer obj)*
der Mouselistener **obj** wird mit allen Recourcen gelöscht.
- j_getmousebutton** *integer function j_getmousebutton (integer mouselistener)*
Liefert die letzte benutzte Mousetaste zurück.
- j_getmousex** *integer function j_getmousex (integer mouselistener)*
Liefert die aktuelle horizontale Position der Mouse zurück.
- j_getmousey** *integer function j_getmousey (integer mouselistener)*
Liefert die aktuelle vertikale Position der Mouse zurück.

Panel

- j_panel** *integer function j_panel (integer obj)*
Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt das Panel **obj** in den Container **cont** ein.
- j_borderpanel** *integer function j_borderpanel (integer obj , integer type)*
Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
- j_button** *integer function j_button (integer obj , character*(*) label)*
Erzeugt einen neuen Button mit dem Textinhalt **label**
- j_canvas** *integer function j_canvas (integer obj , integer width , integer height)*
Erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel.
- j_checkbox** *integer function j_checkbox (integer obj , character*(*) label)*
Erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_choice** *integer function j_choice (integer obj)*
Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an das Panel **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled das Panel **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
das Panel **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled das Panel **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an das Panel **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Panel **obj** in Bildschirm-pixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*

	Liefert die totale Höhe des aktuellen Fonts des Panel obj in Bildschirmpixeln zurück.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Panel obj zurück.
j_getinheight	<i>integer function j_getinheight (integer cont)</i> Liefert die Höhe des Panel obj des inneren Bereichs zurück.
j_getinsets	<i>integer function j_getinsets (integer obj , integer side)</i> Liefert die angeforderte Breite des inneren Randes dem Panel zurück.
j_getinwidth	<i>integer function j_getinwidth (integer cont)</i> Liefert die Breite des Panel obj des inneren Bereichs zurück.
j_getlayoutid	<i>integer function j_getlayoutid (integer obj)</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Panel obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Panel obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Panel obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Panel obj zurück.
j_graphicbutton	<i>integer function j_graphicbutton (integer obj , character*(*) filename)</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>integer function j_graphiclabel (integer obj , character*(*) str)</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt das Panel obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>integer function j_hscrollbar (integer obj)</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an das Panel obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>integer function j_label (integer obj , character*(*) label)</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>integer function j_led (integer obj , integer style , integer color)</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>integer function j_line (integer obj , integer orient , integer style , integer length)</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>integer function j_list (integer obj , integer rows)</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>integer function j_meter (integer obj , character*(*) title)</i> Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung title .
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an das Panel obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (integer obj)</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>integer function j_panel (integer obj)</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Panel .
j_progressbar	<i>integer function j_progressbar (integer obj , integer orient)</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>integer function j_radiogroup (integer obj)</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (integer obj)</i> Gibt alle Elemente aus dem Panel obj frei.

j_release	<i>procedure j_release (integer obj)</i> Gibt das Panel obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>integer function j_scrollpane (integer obj)</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (integer obj , integer align)</i> Setzt die Ausrichtung innerhalb des Panel obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (integer obj)</i> Integriert einen BorderLayout Manager in das Panel obj .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert das Panel obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Panel obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (integer obj)</i> Integriert einen Fixlayout Manager in das Panel obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (integer obj , integer bool)</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Panel obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (integer obj , integer align)</i> Integriert einen Flowlayout Manager in das Panel obj mit der Ausrichtung align .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Panel obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Panel obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Panel obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Panel obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Panel obj auf style .

j_setgridlayout	<i>procedure j_setgridlayout (integer obj , integer row , integer col)</i> Integriert einen Gridlayout Manager in das Panel obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (integer obj , integer hgap)</i> Setzt den horizontalen Abstand innerhalb des Panel obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (integer obj)</i> Entfernt den aktuellen Layout Manager aus dem Panel obj .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Panel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Panel obj auf width and height .
j_setvgap	<i>procedure j_setvgap (integer obj , integer vgap)</i> Setzt den vertikalen Abstand innerhalb des Panel obj auf vgap Pixel.
j_sevensegment	<i>integer function j_sevensegment (integer obj , integer color)</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Panel obj auf dem Bildschirm an.
j_textarea	<i>integer function j_textarea (integer obj , integer rows , integer columns)</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>integer function j_textfield (integer obj , integer columns)</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>integer function j_vscrollbar (integer obj)</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

<h2>PopupMenu</h2>

j_popupmenu	<p><i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.</p>
j_checkmenuitem	<p><i>integer function j_checkmenuitem (integer obj , character*(*) label)</i> Erzeugt ein Checkmenuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_disable	<p><i>procedure j_disable (integer obj)</i> disabled das Popupmenu obj, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
j_dispose	<p><i>procedure j_dispose (integer obj)</i> das Popupmenu obj wird mit allen Recourcen gelöscht.</p>
j_enable	<p><i>procedure j_enable (integer obj)</i> enabled das Popupmenu obj.</p>
j_getlength	<p><i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes das Popupmenu zurück.</p>
j_gettext	<p><i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Popupmenu .</p>
j_menuitem	<p><i>integer function j_menuitem (integer obj , character*(*) label)</i> Erzeugt ein Menuitem mit der Beschriftung label und liefert dessen Eventnummer zurück.</p>
j_seperator	<p><i>procedure j_seperator (integer obj)</i> Erzeugt eine Trennlinie in das Popupmenu .</p>
j_setfontname	<p><i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Popupmenu obj auf name.</p>
j_setfont	<p><i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Popupmenu obj entsprechend der Parameterliste.</p>
j_setfontsize	<p><i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Popupmenu obj auf size.</p>
j_setfontstyle	<p><i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Popupmenu obj auf style.</p>
j_setshortcut	<p><i>procedure j_setshortcut (integer obj , character chr)</i> Bestimmt das Zeichen chr als Shortcut für das Popupmenu .</p>
j_settext	<p><i>procedure j_settext (integer obj , character*(*) str)</i></p>

Ändert den Textinhalt des Popupmenu **obj** auf **str**

j_showpopup

procedure j_showpopup (integer obj , integer xpos , integer ypos)

Zeigt das Popupmenu an der Koordinate (**xpos,ypos**) an.

Printer

- j_printer** *integer function j_printer (integer frame)*
Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
- j_cliprect** *procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)*
Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).
- j_dispose** *procedure j_dispose (integer obj)*
der Printer **obj** wird mit allen Recourcen gelöscht.
- j_drawarc** *procedure j_drawarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
Zeichnet einen Kreisbogen mit dem Mittelpunkt (**x**, **y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_drawcircle** *procedure j_drawcircle (integer obj , integer x , integer y , integer r)*
Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_drawimage** *procedure j_drawimage (integer obj , integer image , integer x , integer y)*
Kopiert das Image mit der Eventnummer **image** an die Position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Die Funktion stellt ein Bild an der Position (**x,y**) mit der Breite **w** und der Höhe **h** dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays **r,g,b** übergeben.
- j_drawline** *procedure j_drawline (integer obj , integer x1 , integer y1 , integer x2 , integer y2)*
Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (**x1,y1**) nach (**x2,y2**)
- j_drawoval** *procedure j_drawoval (integer obj , integer x , integer y , integer rx , integer ry)*
Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_drawpixel** *procedure j_drawpixel (integer obj , integer x , integer y)*
Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (**x,y**).
- j_drawpolygon** *procedure j_drawpolygon (integer obj , integer len , array of integer x , array of integer y)*

- Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawpolyline** *procedure j_drawpolyline (integer obj , integer len , array of integer x , array of integer y)*
Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawrect** *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
Zeichnet ein Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
Zeichnet ein Rechteck mit abgerundeten Ecken an die Position (**x,y**) mit der Breite **width** und der Höhe **height**. Die Radien der Ecken werden durch **arcx** und **arcy** bestimmt.
- j_drawscaledimage** *procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)*
Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter (**sx**, **sy**, **sw**, **sh**) bestimmt wird an die Position (**tx**, **ty**) und scaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**
- j_drawstring** *procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)*
Zeichnet den String **str** an die durch die Parameter (**x,y**) festgelegte Koordinate.
- j_fillarc** *procedure j_fillarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens.
- j_fillcircle** *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_filloval** *procedure j_filloval (integer obj , integer x , integer y , integer rx , integer ry)*
Zeichnet einen gefüllten Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_fillpolygon** *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *procedure j_fillrect (integer obj , integer x , integer y , integer width , integer height)*
Zeichnet ein gefülltes Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.

- j_fillroundrect** *procedure j_fillroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt des Printer .
- j_setxor** *procedure j_setxor (integer obj , integer bool)*
Schaltet den Painting Mode auf XOR, falls bool = .true. ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
- j_translate** *procedure j_translate (integer obj , integer x , integer y)*
Setzt den Ursprung des Koordinatensystems auf **(x, y)**.

Progressbar

- j_progressbar** *integer function j_progressbar (integer obj , integer orient)*
Erzeugt eine Progress Bar mit der Ausrichtung **orient**.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Progressbar **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Progressbar **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Progressbar **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Progressbar **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Progressbar **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Progressbar **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Progressbar **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Progressbar **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Progressbar **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*

Liefert die Länge des **str** des aktuellen Fonts der Progressbar **obj** in Bildschirmpixeln zurück.

- j_getwidth** *integer function j_getwidth (integer obj)*
Liefert die Breite der Progressbar **obj** zurück.
- j_getxpos** *integer function j_getxpos (integer obj)*
Liefert die aktuelle horizontale Position der Progressbar **obj** zurück.
- j_getypos** *integer function j_getypos (integer obj)*
Liefert die aktuelle vertikale Position der Progressbar **obj** zurück.
- j_hide** *procedure j_hide (integer obj)*
Zeigt die Progressbar **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *integer function j_isparent (integer obj , integer cont)*
Liefert `.true.` zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst `.false.`
- j_isvisible** *integer function j_isvisible (integer obj)*
Liefert `.true.` zurück, falls **obj** sichtbar ist, sonst `.false.`
- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an die Progressbar **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an die Progressbar **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt der Progressbar .
- j_release** *procedure j_release (integer obj)*
Gibt die Progressbar **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (integer obj , integer pos)*
Positioniert die Progressbar **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (integer obj , integer r , integer g , , integer b)*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (integer obj , integer r , integer g , , integer b)*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcursor** *integer function j_setcursor (integer obj , integer cursor)*
Setzt den Cursor der Progressbar **obj** auf **cursor**.

j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Progressbar obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Progressbar obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Progressbar obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Progressbar obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Progressbar obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Progressbar obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Progressbar obj auf width and height .
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Progressbar obj auf dem Bildschirm an.

Radiobutton

- j_radiobutton** *integer function j_radiobutton (integer obj , character*(*) label)*
Erzeugt einen Radiobutton mit dem Textinhalt **label** und liefert dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt den Radiobutton **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an den Radiobutton **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled den Radiobutton **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
der Radiobutton **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled den Radiobutton **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an den Radiobutton **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Radiobutton **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts des Radiobutton **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe des Radiobutton **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstate** *integer function j_getstate (integer obj)*

	Liefert <code>.true.</code> , falls der Radiobutton selektiert ist, sonst <code>.false.</code> .
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Radiobutton obj in Bildschirmpixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Radiobutton .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Radiobutton obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Radiobutton obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Radiobutton obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt den Radiobutton obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an den Radiobutton obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an den Radiobutton obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Radiobutton .
j_release	<i>procedure j_release (integer obj)</i> Gibt den Radiobutton obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert den Radiobutton obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Radiobutton obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Radiobutton obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Radiobutton obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Radiobutton obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Radiobutton obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Radiobutton obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt den Radiobutton obj zur angegebenen Koordinate (xpos,ypos) an.
j_setradiogroup	<i>integer function j_setradiogroup (integer rbutton , integer rgroup)</i> Weist dem Radiobutton rbutton die Radiogroup rgroup zu. Der Radiobutton wird aus der alten Radiogroup entfernt.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Radiobutton obj auf width and height .
j_setstate	<i>procedure j_setstate (integer obj , integer bool)</i> Ist bool .true. so wird den Radiobutton selectiert.
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Radiobutton obj auf str
j_show	<i>procedure j_show (integer obj)</i> Zeigt den Radiobutton obj auf dem Bildschirm an.

Sevensegment

- j_sevensegment** *integer function j_sevensegment (integer obj , integer color)*
Erzeugt eine Siebensegment Anzeige der Farbe **color**.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Siebensegment-Anzeige **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Siebensegment-Anzeige **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Siebensegment-Anzeige **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Siebensegment-Anzeige **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Siebensegment-Anzeige **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Siebensegment-Anzeige **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Siebensegment-Anzeige **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*

Liefert die Länge des **str** des aktuellen Fonts der Siebensegment-Anzeige **obj** in Bildschirmpixeln zurück.

- j_getwidth** *integer function j_getwidth (integer obj)*
Liefert die Breite der Siebensegment-Anzeige **obj** zurück.
- j_getxpos** *integer function j_getxpos (integer obj)*
Liefert die aktuelle horizontale Position der Siebensegment-Anzeige **obj** zurück.
- j_getypos** *integer function j_getypos (integer obj)*
Liefert die aktuelle vertikale Position der Siebensegment-Anzeige **obj** zurück.
- j_hide** *procedure j_hide (integer obj)*
Zeigt die Siebensegment-Anzeige **obj** nicht mehr auf dem Bildschirm an.
- j_isparent** *integer function j_isparent (integer obj , integer cont)*
Liefert `.true.` zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst `.false.`
- j_isvisible** *integer function j_isvisible (integer obj)*
Liefert `.true.` zurück, falls **obj** sichtbar ist, sonst `.false.`
- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an die Siebensegment-Anzeige **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt der Siebensegment-Anzeige .
- j_release** *procedure j_release (integer obj)*
Gibt die Siebensegment-Anzeige **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (integer obj , integer pos)*
Positioniert die Siebensegment-Anzeige **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (integer obj , integer r , integer g , integer b)*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (integer obj , integer r , integer g , integer b)*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der Siebensegment-Anzeige obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Siebensegment-Anzeige obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Siebensegment-Anzeige obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Siebensegment-Anzeige obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Siebensegment-Anzeige obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Siebensegment-Anzeige obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Siebensegment-Anzeige obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Siebensegment-Anzeige obj auf width and height .
j_setvalue	<i>procedure j_setvalue (integer obj , integer val)</i> Stellt den aktuellen Wert der Siebensegment-Anzeige obj auf val .
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Siebensegment-Anzeige obj auf dem Bildschirm an.

Scrollpane

- j_scrollpane** *integer function j_scrollpane (integer obj)*
Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Scrollpane **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Scrollpane **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Scrollpane **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Scrollpane **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Scrollpane **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Scrollpane **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Scrollpane **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Scrollpane **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Scrollpane **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*
Liefert die Länge des **str** des aktuellen Fonts der Scrollpane **obj** in Bildschirmpixeln zurück.

- j_getviewportheight** *integer function j_getviewportheight (integer obj)*
Gibt die Höhe des sichtbaren Teils der Scrollpane **obj** zurück.
- j_getviewportwidth** *integer function j_getviewportwidth (integer obj)*
Gibt die Breite des sichtbaren Teils der Scrollpane **obj** zurück.
- j_getwidth** *integer function j_getwidth (integer obj)*
Liefert die Breite der Scrollpane **obj** zurück.
- j_getxpos** *integer function j_getxpos (integer obj)*
Liefert die aktuelle horizontale Position der Scrollpane **obj** zurück.
- j_getypos** *integer function j_getypos (integer obj)*
Liefert die aktuelle vertikale Position der Scrollpane **obj** zurück.
- j_hide** *procedure j_hide (integer obj)*
Zeigt die Scrollpane **obj** nicht mehr auf dem Bildschirm an.
- j_hscrollbar** *integer function j_hscrollbar (integer obj)*
Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
- j_isparent** *integer function j_isparent (integer obj , integer cont)*
Liefert `.true.` zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst `.false.`
- j_isvisible** *integer function j_isvisible (integer obj)*
Liefert `.true.` zurück, falls **obj** sichtbar ist, sonst `.false.`
- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an die Scrollpane **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an die Scrollpane **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt der Scrollpane .
- j_release** *procedure j_release (integer obj)*
Gibt die Scrollpane **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure j_setborderpos (integer obj , integer pos)*
Positioniert die Scrollpane **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (integer obj , integer r , integer g , integer b)*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der Scrollpane obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Scrollpane obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Scrollpane obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Scrollpane obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Scrollpane obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Scrollpane obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Scrollpane obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Scrollpane obj auf width and height .
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Scrollpane obj auf dem Bildschirm an.
j_vscrollbar	<i>integer function j_vscrollbar (integer obj)</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

Textarea

- j_textarea** *integer function j_textarea (integer obj , integer rows , integer columns)*
Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Textarea **obj** in den Container **cont** ein.
- j_appendtext** *procedure j_appendtext (integer obj , character*(*) text)*
Hängt den Text **text** am Ende von **obj** an.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Textarea **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_delete** *procedure j_delete (integer obj , integer start , integer end)*
löscht den Text von Position **start** bis Position **end**.
- j_disable** *procedure j_disable (integer obj)*
disabled die Textarea **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Textarea **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Textarea **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Textarea **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getcolumns** *procedure j_getcolumns (integer obj)*
Gibt die Anzahl der Spalten von **obj** zurück.
- j_getcurpos** *integer function j_getcurpos (integer obj)*
Liefert die Position des Text Cursors zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Textarea **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Textarea **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*

	Liefert die Höhe der Textarea obj zurück.
j_getlength	<i>integer function j_getlength (integer obj)</i> Liefert die Länge des Textinhaltes die Textarea zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getrows	<i>procedure j_getrows (integer obj)</i> Gibt die Anzahl der Reihen von obj zurück.
j_getselend	<i>integer function j_getselend (integer obj)</i> Liefert die Endposition des selectierten Textes.
j_getselstart	<i>integer function j_getselstart (integer obj)</i> Liefert die Anfangsposition des selectierten Textes.
j_getseltext	<i>procedure j_getseltext (integer obj , character*(*) text)</i> Liefert den selectierten Text die Textarea obj zurück.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts der Textarea obj in Bildschirm-pixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt der Textarea .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der Textarea obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der Textarea obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der Textarea obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die Textarea obj nicht mehr auf dem Bildschirm an.
j_inserttext	<i>procedure j_inserttext (integer obj , character*(*) text , integer pos)</i> Fügt zusätzlichen Text text an der Position pos ein.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_isvisible	<i>integer function j_isvisible (integer obj)</i>

Liefert `.true.` zurück, falls **obj** sichtbar ist, sonst `.false.`

- j_keylistener** *integer function j_keylistener (integer obj)*
Bindet an die Textarea **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Bindet an die Textarea **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *integer function j_popupmenu (integer obj , character*(*) label)*
Erzeugt ein neues Popumenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure j_print (integer obj)*
Druckt den Inhalt der Textarea .
- j_release** *procedure j_release (integer obj)*
Gibt die Textarea **obj** aus dem umgebenden Container Element frei.
- j_replacetext** *procedure j_replacetext (integer obj , character*(*) text , integer start , integer end)*
Ersetzt den Text zwischen Position **start** bis **end** durch **text**.
- j_selectall** *procedure j_selectall (integer obj)*
Selectiert den gesamten Textinhalt der Textarea .
- j_selecttext** *procedure j_selecttext (integer obj , integer start , integer end)*
Selectiert den Text von Position **start** bis **end**.
- j_setborderpos** *procedure j_setborderpos (integer obj , integer pos)*
Positioniert die Textarea **obj** an die angegebene Position **pos**. Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
- j_setcolorbg** *procedure j_setcolorbg (integer obj , integer r , integer g , integer b)*
Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
- j_setcolor** *procedure j_setcolor (integer obj , integer r , integer g , integer b)*
Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
- j_setcolumns** *procedure j_setcolumns (integer obj , integer columns)*
Setzt die Anzahl Spalten von **obj** auf **columns**.
- j_setcurpos** *procedure j_setcurpos (integer obj , integer pos)*
Setzt den Text Cursor auf die Position **pos**.
- j_setcursor** *integer function j_setcursor (integer obj , integer cursor)*
Setzt den Cursor der Textarea **obj** auf **cursor**.
- j_seteditable** *procedure j_seteditable (integer obj , integer bool)*
Ist **bool** `.true.` , so ist die Textarea editierbar, ansonsten ist die Textarea nur lesbar für den Benutzer.

j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Textarea obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Textarea obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Textarea obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Textarea obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Textarea obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Textarea obj zur angegebenen Koordinate (xpos,ypos) an.
j_setrows	<i>procedure j_setrows (integer obj , integer rows)</i> Setzt die Anzahl Reihen von obj auf rows .
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Textarea obj auf width and height .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt der Textarea obj auf str
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Textarea obj auf dem Bildschirm an.

Textfield

- j_textfield** *integer function j_textfield (integer obj , integer columns)*
Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt das Textfield **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an das Textfield **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled das Textfield **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
das Textfield **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled das Textfield **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an das Textfield **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getcolumns** *procedure j_getcolumns (integer obj)*
Gibt die Anzahl der Spalten von **obj** zurück.
- j_getcurpos** *integer function j_getcurpos (integer obj)*
Liefert die Position des Text Cursors zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Textfield **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts des Textfield **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe des Textfield **obj** zurück.
- j_getlength** *integer function j_getlength (integer obj)*
Liefert die Länge des Textinhaltes das Textfield zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*

	Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselend	<i>integer function j_getselend (integer obj)</i> Liefert die Endposition des selectierten Textes.
j_getselstart	<i>integer function j_getselstart (integer obj)</i> Liefert die Anfangsposition des selectierten Textes.
j_getseltext	<i>procedure j_getseltext (integer obj , character*(*) text)</i> Liefert den selectierten Text das Textfield obj zurück.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Textfield obj in Bildschirm-pixeln zurück.
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> Liefert den Textinhalt des Textfield .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Textfield obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Textfield obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Textfield obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt das Textfield obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an das Textfield obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an das Textfield obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i>

	Erzeugt ein neues Popumenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Textfield .
j_release	<i>procedure j_release (integer obj)</i> Gibt das Textfield obj aus dem umgebenden Container Element frei.
j_selectall	<i>procedure j_selectall (integer obj)</i> Selectiert den gesamten Textinhalt des Textfield .
j_selecttext	<i>procedure j_selecttext (integer obj , integer start , integer end)</i> Selectiert den Text von Position start bis end .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert das Textfield obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcolumns	<i>procedure j_setcolumns (integer obj , integer columns)</i> Setzt die Anzahl Spalten von obj auf columns .
j_setcurpos	<i>procedure j_setcurpos (integer obj , integer pos)</i> Setzt den Text Cursor auf die Position pos .
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Textfield obj auf cursor .
j_setechochar	<i>procedure j_setechochar (integer obj , character chr)</i> Bestimmt das Zeichen chr als Echo Zeichen für alle Benutzereingaben in das Textfield .
j_seteditable	<i>procedure j_seteditable (integer obj , integer bool)</i> Ist bool .true. , so ist das Textfield editierbar, ansonsten ist das Textfield nur lesbar für den Benutzer.
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Textfield obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Textfield obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Textfield obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Textfield obj auf size .

j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Textfield obj auf style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Textfield obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Textfield obj auf width and height .
j_settext	<i>procedure j_settext (integer obj , character*(*) str)</i> Ändert den Textinhalt des Textfield obj auf str
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Textfield obj auf dem Bildschirm an.

Vscrollbar

- j_vscrollbar** *integer function j_vscrollbar (integer obj)*
Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
- j_add** *procedure j_add (integer obj , integer cont)*
Fügt die Vscrollbar **obj** in den Container **cont** ein.
- j_componentlistener** *integer function j_componentlistener (integer obj , integer kind)*
Bindet an die Vscrollbar **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_disable** *procedure j_disable (integer obj)*
disabled die Vscrollbar **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j_dispose** *procedure j_dispose (integer obj)*
die Vscrollbar **obj** wird mit allen Recourcen gelöscht.
- j_enable** *procedure j_enable (integer obj)*
enabled die Vscrollbar **obj**.
- j_focuslistener** *integer function j_focuslistener (integer obj)*
Bindet an die Vscrollbar **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *integer function j_getfontascent (integer obj)*
Liefert die Oberlänge (Ascent) des aktuellen Fonts der Vscrollbar **obj** in Bildschirmpixeln zurück.
- j_getfontheight** *integer function j_getfontheight (integer obj)*
Liefert die totale Höhe des aktuellen Fonts der Vscrollbar **obj** in Bildschirmpixeln zurück.
- j_getheight** *integer function j_getheight (integer obj)*
Liefert die Höhe der Vscrollbar **obj** zurück.
- j_getparentid** *integer function j_getparentid (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *integer function j_getparent (integer obj)*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getstringwidth** *integer function j_getstringwidth (integer obj , character*(*) str)*
Liefert die Länge des **str** des aktuellen Fonts der Vscrollbar **obj** in Bildschirmpixeln zurück.

j_getvalue	<i>integer function j_getvalue (integer obj)</i> Liefert den aktuellen Wert der Vscrollbar .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite der Vscrollbar obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position der Vscrollbar obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position der Vscrollbar obj zurück.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt die Vscrollbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert .true. zurück, falls das cont die umgebende Komponente von obj ist, sonst .false.
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert .true. zurück, falls obj sichtbar ist, sonst .false.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an die Vscrollbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an die Vscrollbar obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt der Vscrollbar .
j_release	<i>procedure j_release (integer obj)</i> Gibt die Vscrollbar obj aus dem umgebenden Container Element frei.
j_setblockinc	<i>integer function j_setblockinc (integer obj , integer val)</i> Stellt die Blockänderung von der Vscrollbar auf den Wert von val ein.
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert die Vscrollbar obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor der Vscrollbar obj auf cursor .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus der Vscrollbar obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp der Vscrollbar obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font der Vscrollbar obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße der Vscrollbar obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle der Vscrollbar obj auf style .
j_setmax	<i>integer function j_setmax (integer obj , integer val)</i> Stellt die obere Grenze der Vscrollbar auf den Wert von val ein.
j_setmin	<i>integer function j_setmin (integer obj , integer val)</i> Stellt die untere Grenze der Vscrollbar auf den Wert von val ein.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt die Vscrollbar obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe der Vscrollbar obj auf width and height .
j_setslidesize	<i>integer function j_setslidesize (integer obj , integer val)</i> Setzt die Größe des Schiebers auf den Wert von val .
j_setunitinc	<i>integer function j_setunitinc (integer obj , integer val)</i> Stellt die minimale Wertänderung von die Vscrollbar auf den Wert von val ein.
j_setvalue	<i>procedure j_setvalue (integer obj , integer val)</i> Stellt den aktuellen Wert der Vscrollbar obj auf val .
j_show	<i>procedure j_show (integer obj)</i> Zeigt die Vscrollbar obj auf dem Bildschirm an.

Window

j_window	<i>integer function j_window (integer obj)</i> Erzeugt ein neues Window und liefert dessen Eventnummer zurück.
j_add	<i>procedure j_add (integer obj , integer cont)</i> Fügt das Window obj in den Container cont ein.
j_borderpanel	<i>integer function j_borderpanel (integer obj , integer type)</i> Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
j_button	<i>integer function j_button (integer obj , character*(*) label)</i> Erzeugt einen neuen Button mit dem Textinhalt label
j_canvas	<i>integer function j_canvas (integer obj , integer width , integer height)</i> Erzeugt einen neuen Canvas mit der Breite width Pixel und einer Höhe von height Pixel.
j_checkbox	<i>integer function j_checkbox (integer obj , character*(*) label)</i> Erzeugt eine Checkbox mit dem Textinhalt label und liefert dessen Eventnummer zurück.
j_choice	<i>integer function j_choice (integer obj)</i> Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
j_componentlistener	<i>integer function j_componentlistener (integer obj , integer kind)</i> Bindet an das Window obj einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_disable	<i>procedure j_disable (integer obj)</i> disabled das Window obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>procedure j_dispose (integer obj)</i> das Window obj wird mit allen Recourcen gelöscht.
j_enable	<i>procedure j_enable (integer obj)</i> enabled das Window obj .
j_focuslistener	<i>integer function j_focuslistener (integer obj)</i> Bindet an das Window obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
j_getfontascent	<i>integer function j_getfontascent (integer obj)</i> Liefert die Oberlänge (Ascent) des aktuellen Fonts des Window obj in Bildschirmpixeln zurück.
j_getfontheight	<i>integer function j_getfontheight (integer obj)</i>

	Liefert die totale Höhe des aktuellen Fonts des Window obj in Bildschirmpixeln zurück.
j_getheight	<i>integer function j_getheight (integer obj)</i> Liefert die Höhe des Window obj zurück.
j_getinheight	<i>integer function j_getinheight (integer cont)</i> Liefert die Höhe des Window obj des inneren Bereichs zurück.
j_getinsets	<i>integer function j_getinsets (integer obj , integer side)</i> Liefert die angeforderte Breite des inneren Randes dem Window zurück.
j_getinwidth	<i>integer function j_getinwidth (integer cont)</i> Liefert die Breite des Window obj des inneren Bereichs zurück.
j_getlayoutid	<i>integer function j_getlayoutid (integer obj)</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>integer function j_getparent (integer obj)</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Liefert die Länge des str des aktuellen Fonts des Window obj in Bildschirmpixeln zurück.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Liefert die Breite des Window obj zurück.
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Liefert die aktuelle horizontale Position des Window obj zurück.
j_getypos	<i>integer function j_getypos (integer obj)</i> Liefert die aktuelle vertikale Position des Window obj zurück.
j_graphicbutton	<i>integer function j_graphicbutton (integer obj , character*(*) filename)</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>integer function j_graphiclabel (integer obj , character*(*) str)</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>procedure j_hide (integer obj)</i> Zeigt das Window obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>integer function j_hscrollbar (integer obj)</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Bindet an das Window obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>integer function j_label (integer obj , character*(*) label)</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>integer function j_led (integer obj , integer style , integer color)</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>integer function j_line (integer obj , integer orient , integer style , integer length)</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>integer function j_list (integer obj , integer rows)</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>integer function j_meter (integer obj , character*(*) title)</i> Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung title .
j_mouselistener	<i>integer function j_mouselistener (integer obj , integer kind)</i> Bindet an das Window obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>procedure j_pack (integer obj)</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>integer function j_panel (integer obj)</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>integer function j_popupmenu (integer obj , character*(*) label)</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure j_print (integer obj)</i> Druckt den Inhalt des Window .
j_progressbar	<i>integer function j_progressbar (integer obj , integer orient)</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>integer function j_radiogroup (integer obj)</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>procedure j_releaseall (integer obj)</i> Gibt alle Elemente aus dem Window obj frei.

j_release	<i>procedure j_release (integer obj)</i> Gibt das Window obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>integer function j_scrollpane (integer obj)</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>procedure j_setalign (integer obj , integer align)</i> Setzt die Ausrichtung innerhalb des Window obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>procedure j_setborderlayout (integer obj)</i> Integriert einen BorderLayout Manager in das Window obj .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Positioniert das Window obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Setzt den Cursor des Window obj auf cursor .
j_setfixlayout	<i>procedure j_setfixlayout (integer obj)</i> Integriert einen Fixlayout Manager in das Window obj (voreingestellte Layoutmanager).
j_setflowfill	<i>procedure j_setflowfill (integer obj , integer bool)</i> Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von dem Window obj . Ein Flowlayout Manager wird vorausgesetzt.
j_setflowlayout	<i>procedure j_setflowlayout (integer obj , integer align)</i> Integriert einen Flowlayout Manager in das Window obj mit der Ausrichtung align .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Weist den Input Focus dem Window obj zu.
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Setzt den Fonttyp des Window obj auf name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Setzt den Font des Window obj entsprechend der Parameterliste.
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Setzt den Fontgröße des Window obj auf size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Setzt den Fontstyle des Window obj auf style .

j_setgridlayout	<i>procedure j_setgridlayout (integer obj , integer row , integer col)</i> Integriert einen Gridlayout Manager in das Window obj mit row Reihen und col Spalten.
j_sethgap	<i>procedure j_sethgap (integer obj , integer hgap)</i> Setzt den horizontalen Abstand innerhalb des Window obj auf hgap Pixel.
j_setinsets	<i>procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>procedure j_setnolayout (integer obj)</i> Entfernt den aktuellen Layout Manager aus dem Window obj .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Verschiebt das Window obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Setzt Breite und Höhe des Window obj auf width and height .
j_setvgap	<i>procedure j_setvgap (integer obj , integer vgap)</i> Setzt den vertikalen Abstand innerhalb des Window obj auf vgap Pixel.
j_sevensegment	<i>integer function j_sevensegment (integer obj , integer color)</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>procedure j_show (integer obj)</i> Zeigt das Window obj auf dem Bildschirm an.
j_textarea	<i>integer function j_textarea (integer obj , integer rows , integer columns)</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>integer function j_textfield (integer obj , integer columns)</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>integer function j_vscrollbar (integer obj)</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>integer function j_windowlistener (integer window , integer kind)</i> Bindet an das Window window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.

Kapitel 2

Functions

additem

Synopsis procedure **j_additem** (integer obj , character*(*) str)

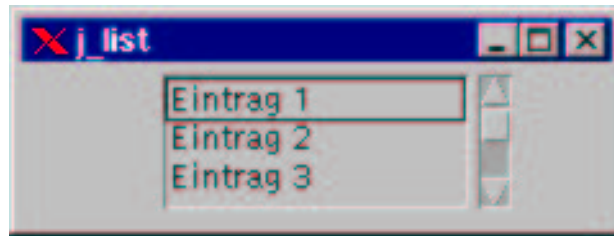
Argumente obj integer
 str character*(*)

Beschreibung Diese Prozedur hängt ein neues Item mit dem Inhalt **str** an die Auswahlliste an. Es wird keine Eventnummer vergeben, statt dessen werden alle Item intern durchnummeriert, beginnend mit 0. Die Zielkomponente erzeugt bei einer Auswahl einen Event. Welches Item ausgewählt wurde, kann mit der Funktion *j_getselect()* ermittelt werden.

Zielkomponenten List, Choice

Beispiel

```
:  
list = j_list(frame,3)  
call j_additem(list,"Eintrag 1")  
call j_additem(list,"Eintrag 2")  
:
```



add

Synopsis	procedure j_add (integer obj , integer cont)
Argumente	obj integer cont integer
Beschreibung	Fügt die Komponente obj in den Container cont ein.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

alertbox

Synopsis procedure **j_alertbox** (integer obj , character*(*) title ,
 character*(*) text , character*(*) button)

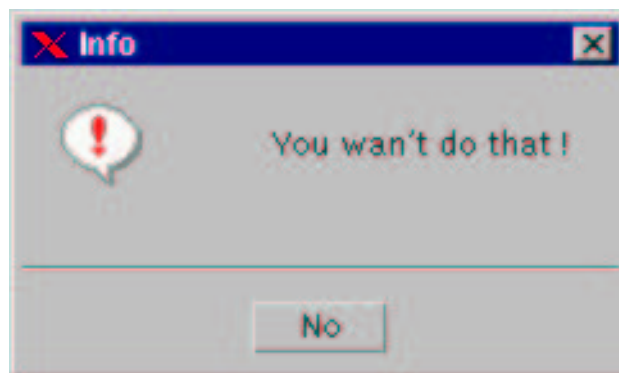
Argumente obj integer
 title character*(*)
 text character*(*)
 button character*(*)

Beschreibung Öffnet eine Alertbox mit der Überschrift **title**, Textinhalt **text** und einem Button. Eine Alertbox ist modal, dh. die Applikation wird solange blockiert, bis der Button oder das Closeicon angeklickt wird. Der Rückgabewert der Funktion beträgt 0 falls das Closeicon geklickt wird, wird der Button geklickt, so wird 1 zurückgegeben.

Zielkomponenten Frame

Beispiel

```
:  
retval = j_alertbox(frame,"Info","You wan't do that !"," No  ")  
:
```



appendtext

Synopsis procedure **j_appendtext** (integer obj , character*(*) text)

Argumente obj integer
 text character*(*)

Beschreibung Hängt den Text **text** am Ende von **obj** an.

Zielkomponenten Textarea

beep

Synopsis procedure **j_beep** ()

Beschreibung Es ertönt ein Systembeep.

borderpanel

Synopsis integer function **j_borderpanel** (integer obj , integer type)

Argumente obj integer
 type integer

Beschreibung Diese Funktion erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Das Panel wird der Komponente zugeordnet, die durch **obj** referenziert wird. Das Borderpanel ist funktionell identisch mit einem normalen Panel, es enthält lediglich einen sichtbaren Rahmen, der die enthaltenen Komponenten auch optisch gruppiert. Der Stil des Rahmens kann aus vier Stiltypen gewählt werden:

- J_LINEDOWN : Der Rahmen besteht aus einer Linie, die optisch abgesenkt erscheint.
- J_LINEUP : Der Rahmen besteht aus einer Linie, die optisch hervorzutreten scheint.
- J_AREADOWN : Das Panel scheint optisch abgesenkt.
- J_AREAUP : Das Panel scheint optisch hervorzutreten.

Ein Borderpanel liefert bei einer Größenänderung einen Event.

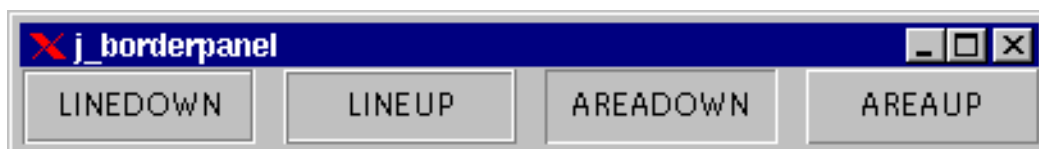
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
call j_setgridlayout(frame,1,4)
p1 = j_borderpanel(frame,J_LINEDOWN)
p2 = j_borderpanel(frame,J_LINEUP)
p3 = j_borderpanel(frame,J_AREADOWN)
p4 = j_borderpanel(frame,J_AREAUP)
:

```



button

Synopsis integer function **j_button** (integer obj , character*(*) label)

Argumente obj integer
 label character*(*)

Beschreibung Die Funktion erzeugt einen neuen Button mit dem Textinhalt **label**. Die Funktion liefert die Eventnummer des Buttons zurück, oder im Fehlerfall -1 . Ein Event wird erzeugt, wenn der Benutzer mit der Maus den Button betätigt.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame = j_frame("j_button")  
button = j_button(frame,"Hello World")  
:
```



canvas

Synopsis integer function **j_canvas** (integer obj , integer width , integer height)

Argumente

obj	integer
width	integer
height	integer

Beschreibung Diese Funktion erzeugt einen neuen Canvas mit der Breite **width** Pixel und einer Höhe von **height** Pixel. Ein Canvas (Leinwand) stellt eine universelle Zeichenfläche zur Verfügung. Die Funktion liefert die Eventnummer des Canvas zurück, oder im Fehlerfall -1. Ein Event wird erzeugt, wenn sich die Größe des Canvas verändert.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
canvas = j_canvas(frame,200,50)
call j_setnamedcolorbg(canvas,J_RED)
:

```



checkbox

Synopsis integer function **j_checkbox** (integer obj , character*(*) label)

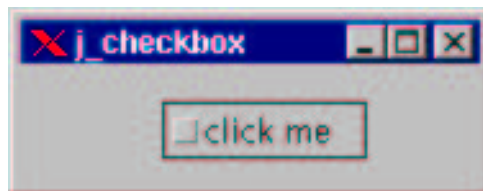
Argumente obj integer
label character*(*)

Beschreibung Diese Funktion erzeugt eine Checkbox mit dem Textinhalt **label** und liefert dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Eine Checkbox ist ein graphisches Element, das zwei Zustände annehmen kann. Sie ist entweder gesetzt oder nicht gesetzt. Checkboxes bestehen intern aus zwei Elementen, einem Label, und einem kleinen graphischen Element, das den Zustand der Checkbox anzeigt. Eine Checkbox liefert immer dann einen Event, wenn sich ihr Zustand ändert.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame = j_frame("j_checkbox")  
checkbox = j_checkbox(frame,"click me")  
:
```



checkmenuitem

Synopsis integer function **j_checkmenuitem** (integer obj ,
 character*(*) label)

Argumente obj integer
 label character*(*)

Beschreibung Die Funktion erzeugt ein Checkmenuitem mit der Beschriftung **label** in der Komponente, die durch **obj** referenziert wird, und liefert dessen Eventnummer zurück. Eine Checkmenuitem erzeugt einen Event, wenn es durch einen Mausklick angewählt wurde. Ein Checkmenuitem kann zwei Zustände annehmen (selected / not selected), die durch ein kleine Icons dargestellt werden. Der Zustand eines Checkmenuitems kann mit der Funktion *j_getstate()* abgefragt werden. Als Zielkomponenten sind nur Menu, Popupmenu und Helpmenu zulässig.

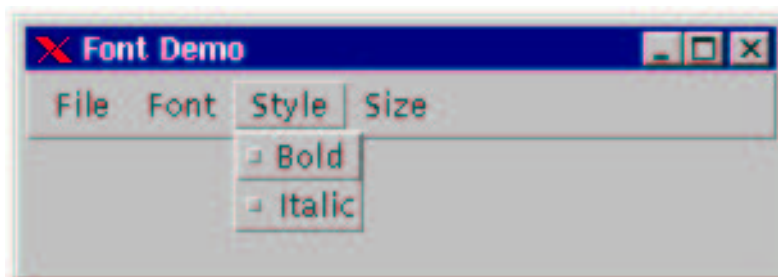
Zielkomponenten Menu, Popupmenu, Helpmenu

Beispiel

```

:
menubar = j_menubar(frame)
:
style = j_menu(menubar,"Style")
bold = j_checkmenuitem(style,"Bold")
italic= j_checkmenuitem(style,"Italic")
:

```



checkbox2

Synopsis

```
procedure j_checkbox2 ( integer obj , character*(*) title ,
character*(*) text , character*(*) button1 , character*(*) button2
)
```

Argumente

obj	integer
title	character*(*)
text	character*(*)
button1	character*(*)
button2	character*(*)

Beschreibung

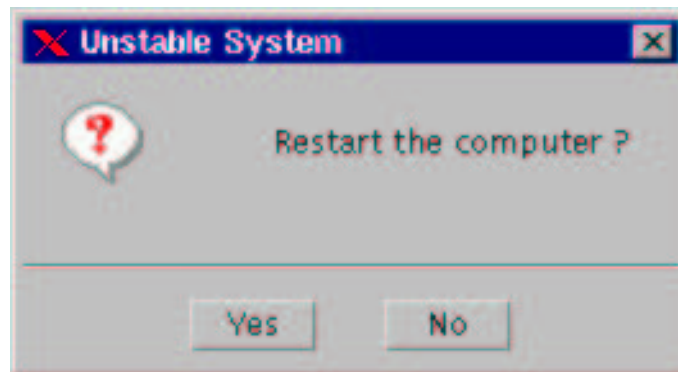
Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und zwei Buttons. Eine Choicebox ist modal, dh. die Applikation wird solange blockiert, bis ein Button oder das Closeicon angeklickt wird. Der Focus ist auf den ersten Button gesetzt. Der Rückgabewert der Funktion beträgt 0 falls das Closeicon geklickt wird, 1 wird Button1 geklickt und 2 bei Button2.

Zielkomponenten

Frame

Beispiel

```
:
retval = j_checkbox2(frame,"Unstable System","Restart the computer ?",
                    "Yes","No")
:
```



checkbox3

Synopsis `procedure j_choicebox3 (integer obj , character*(*) title ,
character*(*) text , character*(*) button1 , character*(*) button2
 , character*(*) button3)`

Argumente

obj	integer
title	character*(*)
text	character*(*)
button1	character*(*)
button2	character*(*)
button3	character*(*)

Beschreibung Öffnet eine Choicebox mit der Überschrift **title**, Textinhalt **text** und drei Buttons. Eine Choicebox ist modal, dh. die Applikation wird solange blockiert, bis ein Button oder das Closeicon angeklickt wird. Der Focus ist auf den ersten Button gesetzt. Der Rückgabewert der Funktion beträgt 0 falls das Closeicon geklickt wird, 1 bei Button1, 2 bei Button2 und 3 bei Button3.

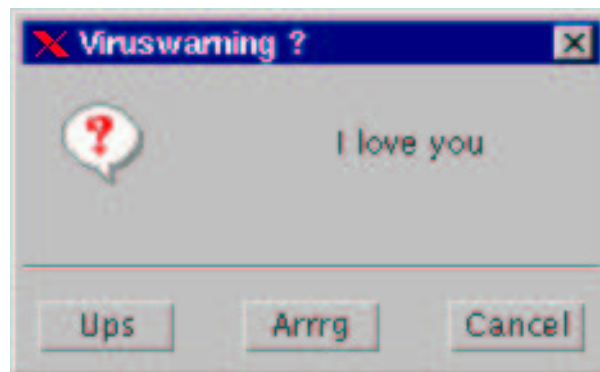
Zielkomponenten Frame

Beispiel

```

:
retval = j_choicebox2(frame,"Viruswarning ?","I love you",
                        "Ups","Arrrg","Cancel")
:

```



choice

Synopsis integer function **j_choice** (integer obj)

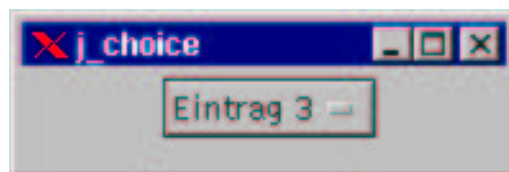
Argumente obj integer

Beschreibung Diese Funktion erzeugt eine neue Choice und gibt deren Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Eine Choice bietet dem Benutzer die Auswahl zwischen mehreren vordefinierten Einträgen. Dabei befindet sich auf der Oberfläche zunächst ein Element das einem Button ähnlich sieht. Wird es angeklickt, so klappt ein Menu heraus, aus dem man ein Element auswählen kann. Wird ein Element ausgewählt, so erzeugt die Choice einen Event.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:
choice = j_choice(frame)
call j_additem(choice,"Eintrag 1")
call j_additem(choice,"Eintrag 2")
:
```



cliprect

Synopsis	procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)
Argumente	obj integer x integer y integer width integer height integer
Beschreibung	Setzt den Clipping Bereich auf das spezifizierte Rechteck (x , y , width , height).
Zielkomponenten	Canvas, Image, Printer

componentlistener

Synopsis	integer function j_componentlistener (integer obj , integer kind)
Argumente	obj integer kind integer
Beschreibung	Bindet an die Komponente obj einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für kind sind: <ul style="list-style-type: none">• J.RESIZED : Der Componentlistener liefert einen Event, wenn die Komponente obj die Größe ändert.• J.HIDDEN : Der Componentlistener liefert einen Event, wenn die Komponente obj verdeckt wird.• J.SHOWN : Der Componentlistener liefert einen Event, wenn die Komponente obj sichtbar wird.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

connect

Synopsis integer function **j_connect** (character*(*) hostname)

Argumente hostname character*(*)

Beschreibung Diese Funktion stellt eine Verbindung zu einem laufenden JAPI Kernel her. Der Kernel wird auf dem Rechner **hostname** gesucht. **hostname** kann einen Namen oder eine IP Adresse beinhalten. Auf diese Weise kann eine Ausgabeumlenkung erfolgen, dh. Applikation und Oberfläche laufen auf verschiedenen Hosts. Schlägt die Verbindung fehl, so wird als Fehlercode `.false.` zurückgegeben

Beispiel

```
:
if( .not. j_connect("atan.japi.de")) then

    or

if( .not. j_connect("127.0.0.1")) then
:
```

delete

Synopsis procedure **j_delete** (integer obj , integer start , integer end)

Argumente obj integer
 start integer
 end integer

Beschreibung löscht den Text von Position **start** bis Position **end**.

Zielkomponenten Textarea

deselect

Synopsis integer function **j_deselect** (integer obj , integer item)

Argumente obj integer
 item integer

Beschreibung Deselectiert das Item mit dem Index **item**.

Zielkomponenten List

dialog

Synopsis integer function **j_dialog** (integer obj , character*(*) label)

Argumente obj integer
 label character*(*)

Beschreibung Dies Funktion erzeugt ein neues Dialogwindow mit dem Titel **label** und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgeliefert. Ein Dialogfenster ist ein eigenständiges Fenster, ebenso wie ein Frame. Allerdings kann ein Dialogfenster keine Menüleiste besitzen. Als Zielobjekt dieser Funktion ist nur ein Frame zulässig. Ein Dialog wird erst angezeigt, wenn ein entsprechendes *j_show()* Kommando ausgeführt wurde. Ein Dialog erzeugt einen Event, wenn das Close Icon im Fensterrahmen angeklickt wird.

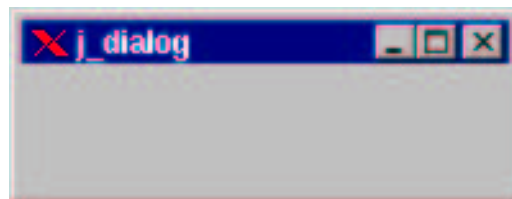
Zielkomponenten Frame

Beispiel

```

:
dialog = j_dialog(frame,"j_dialog")
call j_setsize(dialog,200,80)
call j_show(dialog)
:

```



disable

Synopsis	procedure j_disable (integer obj)
Argumente	obj integer
Beschreibung	disabled die Komponente obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

dispose

Synopsis	procedure j_dispose (integer obj)
Argumente	obj integer
Beschreibung	die Komponente obj wird mit allen Recourcen gelöscht.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Canvas, Image, Printer, Keylistener, Focuslistener, Mousetlistener

drawarc

Synopsis procedure **j_drawarc** (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)

Argumente

obj	integer
x	integer
y	integer
rx	integer
ry	integer
arc1	integer
arc2	integer

Beschreibung Zeichnet einen Kreisbogen mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens. Bezugswert ist die Horizontale, von der die Winkel entgegen des Uhrzeigersinns abgetragen werden. Die Winkel sind im Bogenmaß anzugeben. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

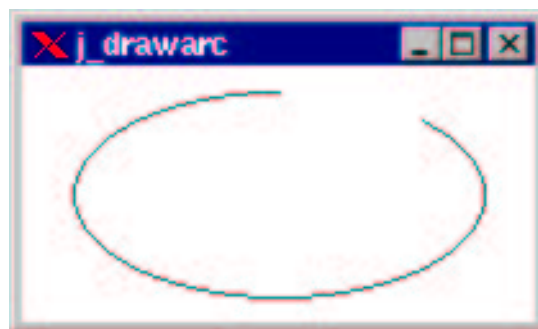
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,200,100)
call j_drawarc(canvas,100,50,80,40,45,-270)
:

```



drawcircle

Synopsis procedure **j_drawcircle** (integer obj , integer x , integer y ,
integer r)

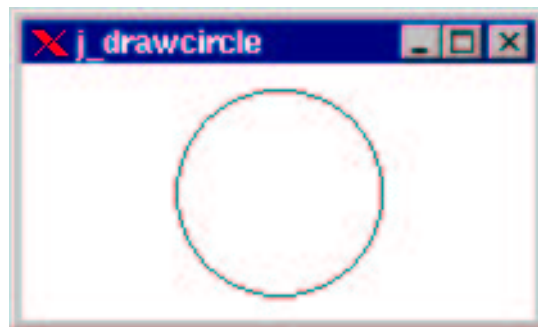
Argumente obj integer
 x integer
 y integer
 r integer

Beschreibung Zeichnet einen Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas = j_canvas(frame,200,100)  
call j_drawcircle(canvas,100,50,40)  
:
```



drawimagesource

Synopsis	procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)
Argumente	obj integer x integer y integer w integer h integer r array of integer g array of integer b array of integer
Beschreibung	Die Funktion stellt ein Bild an der Position (x,y) mit der Breite w und der Höhe h dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays r,g,b übergeben.
Zielkomponenten	Canvas, Image, Printer

drawimage

Synopsis	procedure j_drawimage (integer obj , integer image , integer x , integer y)
Argumente	obj integer image integer x integer y integer
Beschreibung	Kopiert das Image mit der Eventnummer image an die Position (x , y).
Zielkomponenten	Canvas, Image, Printer

drawline

Synopsis procedure **j_drawline** (integer obj , integer x1 , integer y1 ,
integer x2 , integer y2)

Argumente obj integer
 x1 integer
 y1 integer
 x2 integer
 y2 integer

Beschreibung Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von Position (**x1,y1**) nach Position (**x2,y2**). Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,256,50)
call j_drawline(canvas,0,0,256,50)
:

```



drawoval

Synopsis procedure **j_drawoval** (integer obj , integer x , integer y ,
integer rx , integer ry)

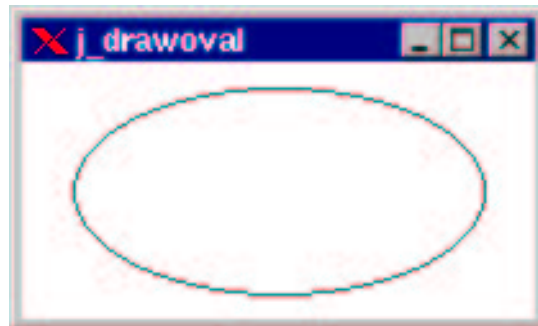
Argumente obj integer
 x integer
 y integer
 rx integer
 ry integer

Beschreibung Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Haupttra-
dien **rx** und **ry** in der momentan gesetzten Vordergrundfarbe.
Nullpunkt des Koordinatensystems ist die linke obere Ecke der
Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas = j_canvas(frame,200,100)  
call j_drawoval(canvas,100,50,80,40)  
:
```



drawpixel

Synopsis `procedure j_drawpixel (integer obj , integer x , integer y)`

Argumente `obj integer`
 `x integer`
 `y integer`

Beschreibung Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position **(x,y)**. Bezugspunkt ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

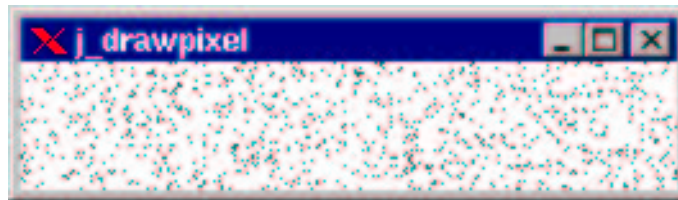
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,256,50)
do i=0,1000
    call j_drawpixel(canvas, mod(j_random(),256), mod(j_random(),50))
end do
:

```



drawpolygon

Synopsis procedure **j_drawpolygon** (integer obj , integer len , array of integer x , array of integer y)

Argumente obj integer
 len integer
 x array of integer
 y array of integer

Beschreibung Zeichnet eine Polylinie in der momentan gesetzten Vordergrundfarbe durch die Punkte, die durch die Arrays **x** und **y** gegeben sind. Beide Arrays müssen die mindestens **len** Elemente besitzen. Das Polygon wird automatisch geschlossen. Daher müssen die erste und letzte Koordinate nicht übereinstimmen. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

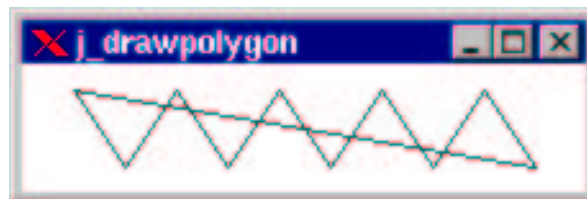
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
data x /20,40,60,80,100,120,140,160,180,200/
data y /10,40,10,40,10,40,10,40,10,40/
canvas = j_canvas(frame,256,50)
call j_drawpolygon(canvas,10,x,y)
:

```



drawpolyline

Synopsis procedure **j_drawpolyline** (integer obj , integer len , array of integer x , array of integer y)

Argumente

obj	integer
len	integer
x	array of integer
y	array of integer

Beschreibung Zeichnet eine Polylinie in der momentan gesetzten Vordergrundfarbe durch die Punkte, die durch die Arrays **x** und **y** gegeben sind. Beide Arrays müssen die mindestens **len** Elemente besitzen. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

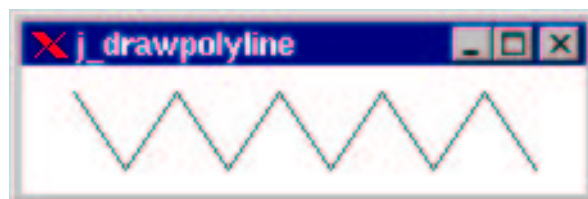
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
data x /20,40,60,80,100,120,140,160,180,200/
data y /10,40,10,40,10,40,10,40,10,40/
canvas = j_canvas(frame,256,50)
call j_drawpolyline(canvas,10,x,y)
:

```



drawrect

Synopsis procedure **j_drawrect** (integer obj , integer x , integer y ,
integer width , integer height)

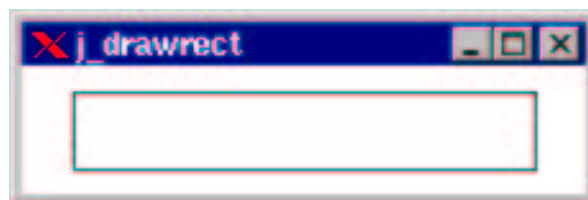
Argumente obj integer
 x integer
 y integer
 width integer
 height integer

Beschreibung Zeichnet ein Rechteck in der momentan gesetzten Vordergrundfarbe an die Position (**x,y**) mit der Breite **width** und der Höhe **height**. Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas = j_canvas(frame,220,50)  
call j_drawrect(canvas,20,10,180,30)  
:
```



drawroundrect

Synopsis procedure **j_drawroundrect** (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)

Argumente

obj	integer
x	integer
y	integer
width	integer
height	integer
arcx	integer
arcy	integer

Beschreibung

Zeichnet ein Rechteck mit abgerundeten Ecken an die Position **(x,y)**. Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt. Diese Parameter bezeichnen die Radien der Rundungen in horizontaler bzw. vertikaler Richtung.

Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

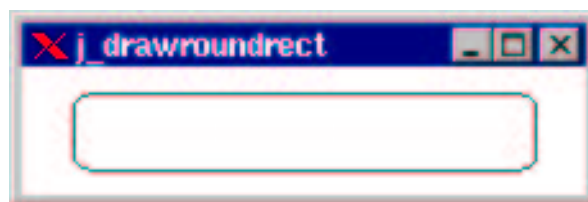
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,220,50)
call j_drawroundrect(canvas,20,10,180,30,10,5)
:

```



drawscaledimage

Synopsis	procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)
Argumente	obj integer image integer sx integer sy integer sw integer sh integer tx integer ty integer tw integer th integer
Beschreibung	Kopiert einen Ausschnitt des Bildes image , der durch die Parameter (sx , sy , sw , sh) bestimmt wird an die Position (tx , ty) und scaliert den Ausschnitt auf die Breite tw und die Höhe th
Zielkomponenten	Canvas, Image, Printer

drawstring

Synopsis procedure **j_drawstring** (integer obj , integer x , integer y ,
 character*(*) str)

Argumente

obj	integer
x	integer
y	integer
str	character*(*)

Beschreibung Die Prozedur zeichnet einen String an die Position (**x,y**). Die durch **obj** referenzierte Komponente muss vom Typ Drawable sein (siehe Zielkomponenten). Die übergebenen Koordinaten beziehen sich auf die linke untere Ecke des Rechtecks, welches sich gedanklich um den gezeichneten String legen läßt.

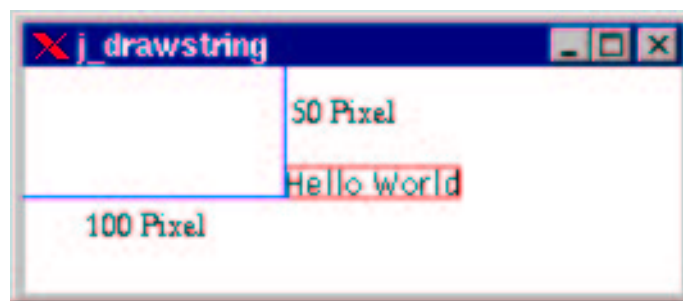
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
call j_drawstring(canvas,100,50,"Hello World")
:

```



enable

Synopsis	procedure j_enable (integer obj)
Argumente	obj integer
Beschreibung	enabled die Komponente obj .
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

filedialog

Synopsis procedure **j_filedialog** (integer frame , character*(*) title ,
character*(*) directory , character*(*) filename)

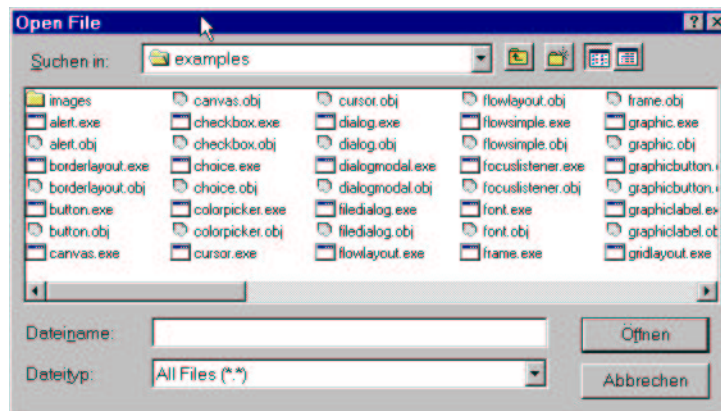
Argumente frame integer
 title character*(*)
 directory character*(*)
 filename character*(*)

Beschreibung Öffnet eine Filedialogbox in dem angegebenen **directory** mit dem
Title **title** und liefert den ausgewählten Dateiname in **filename**
zurück. Ist `"/S"` im Titel enthalten wird die SAVE-Filedialogbox
aufgerufen. Der Teilstring `"/S"` wird aus dem Titelstring entfernt.

Zielkomponenten Frame

Beispiel

```
:
call j_filedialog(frame,"Save/S File","..",filename)
:
```



fileselect

Synopsis procedure **j_fileselect** (integer frame , character*(*) title ,
 character*(*) filter , character*(*) filename)

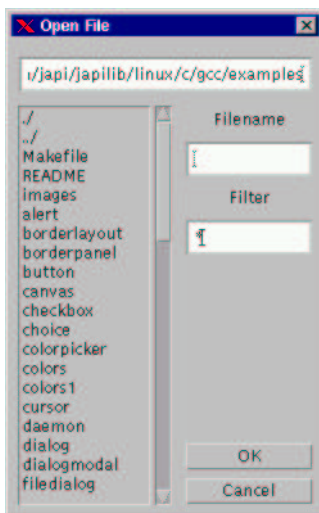
Argumente frame integer
 title character*(*)
 filter character*(*)
 filename character*(*)

Beschreibung Öffnet eine Fileselectorbox mit dem voreingestellten **filename** und dem Title **title** und liefert den ausgewählten Dateiname in **filename** zurück. Im Gegensatz zum Filedialog kann ein **filter** zur Dateiauswahl angegeben werden. Da der Fileselector auf Applikationsebene realisiert ist, kann er auch bei einer Ausgabeumlenkung verwendet werden.

Zielkomponenten Frame

Beispiel

```
:
call j_fileselect(frame,"Open File","*",filename)
:
```



fillarc

Synopsis procedure **j_fillarc** (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)

Argumente

obj	integer
x	integer
y	integer
rx	integer
ry	integer
arc1	integer
arc2	integer

Beschreibung

Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt **(x,y)** und den Haupttradien **rx** und **ry**. Die Parameter **arc1** und **arc2** bestimmen den Start- bzw. den Endwinkel des Kreisbogens. Bezugswert ist die Horizontale von der die Winkel entgegen des Uhrzeigersinns abgetragen werden. Die Winkel sind im Bogenmaß anzugeben.

Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponente die durch **obj** referenziert wird.

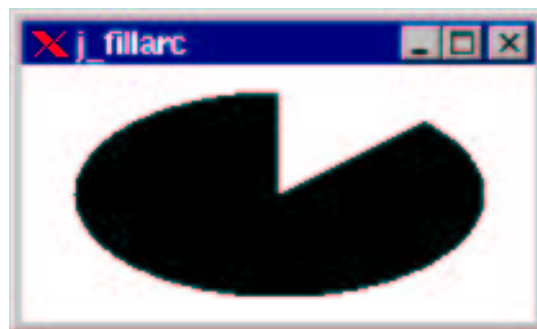
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,200,100)
call j_fillarc(canvas,100,50,80,40,45,-270)
:

```



fillcircle

Synopsis procedure **j_fillcircle** (integer obj , integer x , integer y , integer r)

Argumente obj integer
 x integer
 y integer
 r integer

Beschreibung Zeichnet einen gefülltes Kreis mit dem Mittelpunkt **(x,y)** und dem Radius **r**. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:  
canvas = j_canvas(frame,200,100)  
call j_fillcircle(canvas,100,50,40)  
:
```



filloval

Synopsis `procedure j_filloval (integer obj , integer x , integer y , integer rx , integer ry)`

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>rx</code>	<code>integer</code>
<code>ry</code>	<code>integer</code>

Beschreibung Zeichnet einen gefüllten Oval mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry** in der momentan gesetzten Vordergrundfarbe. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,200,100)
call j_filloval(canvas,100,50,80,40)
:

```



fillpolygon

Synopsis procedure **j_fillpolygon** (integer obj , integer len , array of integer x , array of integer y)

Argumente

obj	integer
len	integer
x	array of integer
y	array of integer

Beschreibung Zeichnet eine gefülltes Polygon in der momentan gesetzten Vordergrundfarbe durch die Punkte, die durch die Arrays **x** und **y** gegeben sind. Beide Arrays müssen die mindestens **len** Elemente besitzen. Das Polygon wird automatisch geschlossen. Daher müssen die erste und letzte Koordinate nicht übereinstimmen. Eingeschlossene Bereiche werden gefüllt dargestellt. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponente die durch **obj** referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
data x /20,40,60,80,100,120,140,160,180,200/
data y /10,40,10,40,10,40,10,40,10,40/
canvas = j_canvas(frame,256,50)
call j_fillpolygon(canvas,10,x,y)
:

```



fillrect

Synopsis procedure **j_fillrect** (integer obj , integer x , integer y , integer width , integer height)

Argumente

obj	integer
x	integer
y	integer
width	integer
height	integer

Beschreibung Zeichnet eine gefülltes Rechteck in der momentan gesetzten Vordergrundfarbe an die Position **(x,y)** mit der Breite **width** und der Höhe **height**. Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

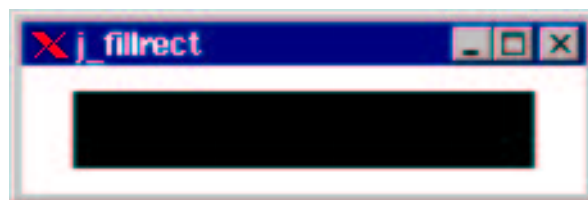
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,220,50)
call j_fillrect(canvas,20,10,180,30)
:

```



fillroundrect

Synopsis `procedure j_fillroundrect (integer obj , integer x , integer y
, integer width , integer height , integer arcx , integer arcy)`

Argumente	obj integer
	x integer
	y integer
	width integer
	height integer
	arcx integer
	arcy integer

Beschreibung Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position (**x,y**). Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt. Diese Parameter bezeichnen die Radien der Rundungen in horizontaler bzw. vertikaler Richtung.
 Bezugspunkt des Rechtecks ist die linke obere Ecke. Nullpunkt des Koordinatensystems ist die linke obere Ecke der Komponenten, die durch **obj** referenziert wird.

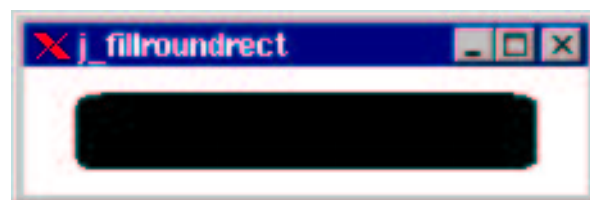
Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,220,50)
call j_fillroundrect(canvas,20,10,180,30,10,5)
:

```



focuslistener

Synopsis	integer function j_focuslistener (integer obj)
Argumente	obj integer
Beschreibung	Bindet an die Komponente obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

frame

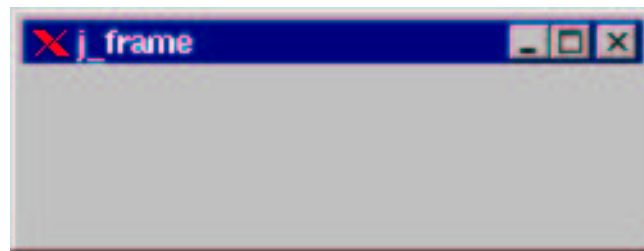
Synopsis integer function **j_frame** (character*(*) label)

Argumente label character*(*)

Beschreibung Diese Funktion erzeugt einen Frame mit der Titelzeile **label**. Ein Frame ist *das* elementare Fenster einer Applikation, an die die anderen graphischen Elemente gebunden werden. Die Funktion liefert die Eventnummer des Frames zurück. Ein Event wird erzeugt, indem der Benutzer das Close Icon im Fensterrahmen anklickt. Ein Frame wird zunächst nur erzeugt, aber nicht angezeigt. Erst nach dem Aufruf der Prozedur *j_show(frame)* wird der Frame sichtbar.

Beispiel

```
:  
frame = j_frame("j_frame")  
call j_show(frame)  
:
```



getaction

Synopsis integer function **j_getaction** ()

Beschreibung Liefert den nächsten Benutzerevent, oder 0 falls kein weiterer Event vorliegt.

getcolumns

Synopsis procedure **j_getcolumns** (integer obj)

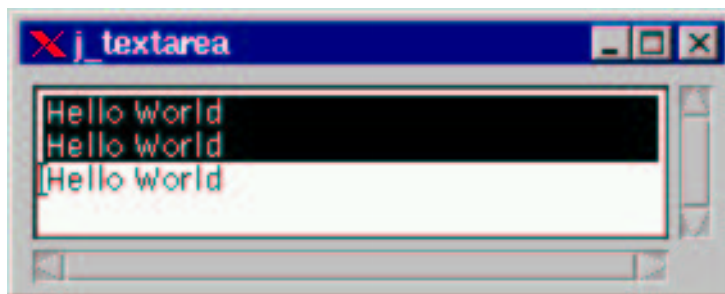
Argumente obj integer

Beschreibung Gibt die Anzahl der Spalten von **obj** zurück.

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

```
:  
text = j_text(frame,30,4)  
call j_getcolumns(text)  
:  
> 30
```



getcurpos

Synopsis integer function **j_getcurpos** (integer obj)

Argumente obj integer

Beschreibung Liefert die Position des Text Cursors zurück.

Zielkomponenten Textarea, Textfield

getdanger

Synopsis procedure **j_getdanger** (integer obj)

Argumente obj integer

Beschreibung Liefert den Warnlevel der Komponenten **obj** zurück.

Zielkomponenten Meter

getfontascent

Synopsis	integer function j_getfontascent (integer obj)
Argumente	obj integer
Beschreibung	Liefert die Oberlänge (Ascent) des aktuellen Fonts der Komponenten obj in Bildschirmpixeln zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

getfontheight

Synopsis	integer function j_getfontheight (integer obj)
Argumente	obj integer
Beschreibung	Liefert die totale Höhe des aktuellen Fonts der Komponenten obj in Bildschirmpixeln zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

getheight

Synopsis	integer function j_getheight (integer obj)
Argumente	obj integer
Beschreibung	Die Funktion liefert die Höhe der Komponenten obj zurück. Neben den graphischen Elementen kann auch die Höhe eines Image ermittelt werden. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Image
Beispiel	<pre>: label = j_getlabel(frame,"Hello World") write(*,*) j_getheight(label) : > 22</pre>

getimagesource

Synopsis	integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)
Argumente	obj integer x integer y integer w integer h integer r array of integer g array of integer b array of integer
Beschreibung	Die Funktion liefert einen Ausschnitt vom Punkt (x , y) der Breite w und der Höhe h zurück. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays r,g,b abgelegt.
Zielkomponenten	Canvas, Image

getimage

Synopsis	integer function j_getimage (integer obj)
Argumente	obj integer
Beschreibung	Kopiert den Inhalt der Komponenten obj in ein Image und liefert dessen Eventnummer zurück.
Zielkomponenten	Canvas, Image

getinheight

Synopsis	integer function j_getinheight (integer cont)
Argumente	cont integer
Beschreibung	Die Funktion liefert die Höhe des inneren Bereichs der Komponenten obj zurück. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre>: frame = j_frame("Hello World") call j_setsize(frame,300,400) write(*,*) j_getinheight(frame) : > 370</pre>

getinsets

Synopsis integer function **j_getinsets** (integer obj , integer side)

Argumente obj integer
 side integer

Beschreibung Liefert die angeforderte Breite des inneren Randes der Komponenten zurück. Der Parameter **side** kann folgende Werte annehmen:

- J_TOP: die Höhe des oberen Rands wird zurueckgeliefert.
- J_BOTTOM: die Höhe des unteren Rands wird zurueckgeliefert.
- J_LEFT: die Breite des linken Rands wird zurueckgeliefert.
- J_RIGHT: die Breite des rechten Rands wird zurueckgeliefert.

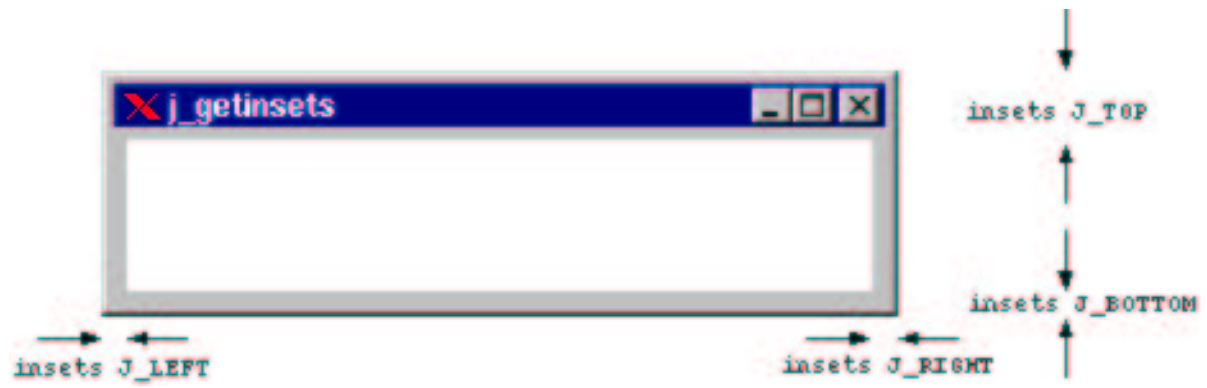
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame("j_getinsets")
write (*,*) j_getinsets(frame,J_TOP),j_getinsets(frame,J_BOTTOM),
           j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT)
:
> 25 5 5 6

```



getinwidth

Synopsis	integer function j_getinwidth (integer cont)
Argumente	cont integer
Beschreibung	Die Funktion liefert die Breite des inneren Bereich der Komponenten obj zurück. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre>: frame = j_frame("Hello World") call j_setsize(frame,300,400) write(*,*) j_getinwidth(frame) : > 289</pre>

getitemcount

Synopsis integer function **j_getitemcount** (integer obj)

Argumente obj integer

Beschreibung Liefert die Anzahl der Items der Komponenten **obj** zurück.

Zielkomponenten List, Choice

getitem

Synopsis	procedure j_getitem (integer obj , integer item , character*(*) str)
Argumente	obj integer item integer str character*(*)
Beschreibung	liefert den Inhalt des Items mit dem Index item zurück.
Zielkomponenten	List, Choice

getkeychar

Synopsis	integer function j_getkeychar (integer obj)
Argumente	obj integer
Beschreibung	Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.
Zielkomponenten	KeyListener

getkeycode

Synopsis	integer function j_getkeycode (integer obj)
Argumente	obj integer
Beschreibung	Liefert den Keycode, der zuletzt gedrückten Taste zurück.
Zielkomponenten	KeyListener

getlayoutid

Synopsis	integer function j_getlayoutid (integer obj)
Argumente	obj integer
Beschreibung	Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	

```
:  
call j_setgridlayout(frame,2,2)  
grid = j_getlayoutid(frame)  
:
```

getlength

Synopsis	integer function j_getlength (integer obj)
Argumente	obj integer
Beschreibung	Liefert die Länge des Textinhaltes die Komponente zurück.
Zielkomponenten	Textarea, Textfield, Dialog, Frame, Button, MenuItem, CheckBox-MenuItem, Menu, HelpMenu, Popupmenu

getmousebutton

Synopsis	integer function j_getmousebutton (integer mousetaster)
Argumente	mousetasterinteger
Beschreibung	Liefert die letzte gedrückte Mauseaste zurück. Der Rückgabewert ist <ul style="list-style-type: none">• J_LEFT linke Mauseaste• J_CENTER mittlere Mauseaste• J_RIGHT rechte Mauseaste
Zielkomponenten	Mousetaster

getmousex

Synopsis integer function **j_getmousex** (integer mouset listener)

Argumente mouset listenerinteger

Beschreibung Liefert die aktuelle horizontale Position der Mouse zurück.

Zielkomponenten Mouset listener

getmousey

Synopsis integer function **j_getmousey** (integer mouset listener)

Argumente mouset listenerinteger

Beschreibung Liefert die aktuelle vertikale Position der Mouse zurück.

Zielkomponenten Mouset listener

getparentid

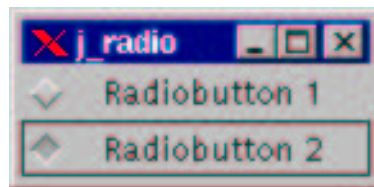
Synopsis	integer function j_getparentid (integer obj)
Argumente	obj integer
Beschreibung	Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Beispiel

```

:
radio1   = j_radiobutton(j_radiogroup(frame),"Radiobutton 1")
radio2   = j_radiobutton(j_getparentid(radio1),"Radiobutton 2")
:

```



getparent

Synopsis integer function **j_getparent** (integer obj)

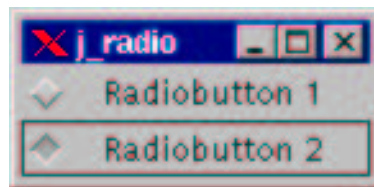
Argumente obj integer

Beschreibung Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Beispiel

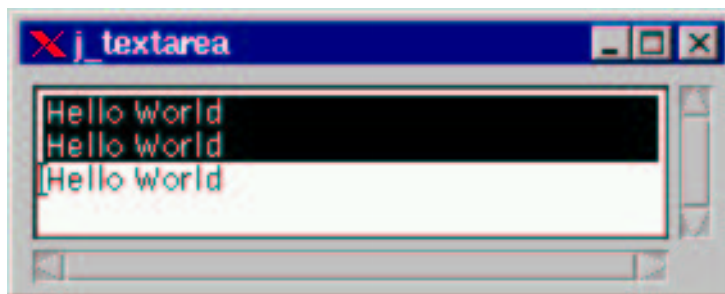
```
:  
radio1 = j_radiobutton(j_radiogroup(frame),"Radiobutton 1")  
radio2 = j_radiobutton(j_getparent(radio1),"Radiobutton 2")  
:
```



getrows

Synopsis	procedure j_getrows (integer obj)
Argumente	obj integer
Beschreibung	Gibt die Anzahl der Reihen von obj zurck.
Zielkomponenten	Textarea, Gridlayout
Beispiel	

```
:  
text = j_text(frame,30,4)  
call j_getrows(text)  
:  
> 4
```



getscaledimage

Synopsis	integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)
Argumente	obj integer x integer y integer sw integer sh integer tw integer th integer
Beschreibung	Kopiert den Inhalt des Ausschnitts von Positio (x , y) it der Breite sw und der höhe sh in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von tw und einer Höhe von th skaliert.
Zielkomponenten	Canvas, Image

getscreenheight

Synopsis	integer function j_getscreenheight ()
Beschreibung	Liefert die Höhe des Bildschirms in Pixeln zurück. Ist ein virtueller Bildschirm installiert, so wird dessen Höhe zurückgeliefert.
Beispiel	<pre>: write(*,*) j_getscreenwidth(), j_getscreenheight() : > 1280 1024</pre>

getscreenwidth

Synopsis integer function **j_getscreenwidth** ()

Beschreibung Liefert die Breite des Bildschirms in Pixeln zurück. Ist ein virtueller Bildschirm installiert, so wird dessen Breite zurückgeliefert.

Beispiel

```
:  
write(*,*) j_getscreenwidth(), j_getscreenheight()  
:  
  
> 1280 1024
```

getselect

Synopsis	integer function j_getselect (integer obj)
Argumente	obj integer
Beschreibung	Liefert den Index des selectierten Items der Komponenten .
Zielkomponenten	List, Choice

getselend

Synopsis integer function **j_getselend** (integer obj)

Argumente obj integer

Beschreibung Liefert die Endposition des selectierten Textes.

Zielkomponenten Textarea, Textfield

getselstart

Synopsis	integer function j_getselstart (integer obj)
Argumente	obj integer
Beschreibung	Liefert die Anfangsposition des selectierten Textes.
Zielkomponenten	Textarea, Textfield

getseltext

Synopsis procedure **j_getseltext** (integer obj , character*(*) text)

Argumente obj integer
 text character*(*)

Beschreibung Liefert den selectierten Text die Komponente **obj** zurück.

Zielkomponenten Textarea, Textfield

getstate

Synopsis	integer function j_getstate (integer obj)
Argumente	obj integer
Beschreibung	Liefert <code>.true.</code> , falls die Komponente selektiert ist, sonst <code>.false.</code> .
Zielkomponenten	Checkbox, Radiobutton, CheckmenuItem, Led

getstringwidth

Synopsis	integer function j_getstringwidth (integer obj , character*(*) str)
Argumente	obj integer str character*(*)
Beschreibung	Liefert die Länge des str des aktuellen Fonts der Komponenten obj in Bildschirmpixeln zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

gettext

Synopsis	procedure j_gettext (integer obj , character*(*) str)
Argumente	obj integer str character*(*)
Beschreibung	Liefert den Textinhalt der, durch obj referenzierten, Komponenten im Parameter str zurück. In C und in Pascal wird ein Pointer auf str zurückgeliefert. Der übergebene String muß groß genug sein, um den gesamten Inhalt des Textes aufzunehmen. Ist die Komponente vom Typ Frame oder Dialog, so wird der Titelstring des Fensters zurückgegeben. Ansonsten wird der Textinhalt der Komponenten zurückgeliefert.
Zielkomponenten	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield
Beispiel	<pre> character*256 str : label = j_label(frame,"Hello World") call j_gettext(label,str) write(*,*) str : > Hello World </pre>

getvalue

Synopsis integer function **j_getvalue** (integer obj)

Argumente obj integer

Beschreibung Liefert den aktuellen Wert der Komponenten .

Zielkomponenten Scrollbar

getviewportheight

Synopsis	integer function j_getviewportheight (integer obj)
Argumente	obj integer
Beschreibung	Gibt die Höhe des sichtbaren Teils der Komponenten obj zurück.
Zielkomponenten	Scrollpane

getviewportwidth

Synopsis integer function **j_getviewportwidth** (integer obj)

Argumente obj integer

Beschreibung Gibt die Breite des sichtbaren Teils der Komponenten **obj** zurück.

Zielkomponenten Scrollpane

getwidth

Synopsis	integer function j_getwidth (integer obj)
Argumente	obj integer
Beschreibung	Die Funktion liefert die Breite der Komponenten obj zurück. Neben den graphischen Elementen kann auch die Breite eines Image ermittelt werden. Die Maßeinheit sind Bildschirmpixel.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Image
Beispiel	<pre>: label = j_getlabel(frame,"Hello World") write(*,*) j_getwidth(label) : > 84</pre>

getxpos

Synopsis	integer function j_getxpos (integer obj)
Argumente	obj integer
Beschreibung	Liefert die aktuelle horizontale Position der Komponenten obj zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

getypos

Synopsis	integer function j_getypos (integer obj)
Argumente	obj integer
Beschreibung	Liefert die aktuelle vertikale Position der Komponenten obj zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

graphicbutton

Synopsis integer function **j_graphicbutton** (integer obj , character*(*)
filename)

Argumente obj integer
 filename character*(*)

Beschreibung Die Funktion erzeugt einen neuen Button mit einem Icon als Inhalt. Die Datei, die das Icon enthält, ist im Parameter **filename** anzugeben. Das Icon muß im GIF oder JPEG-Format vorliegen. Die Funktion liefert die Eventnummer des Buttons zurück, oder im Fehlerfall -1 . Ein Event wird erzeugt, wenn der Benutzer mit der Maus den Button betätigt.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame = j_frame("j_graphicbutton")  
button = j_graphicbutton(frame,"save.gif")  
:
```



graphiclabel

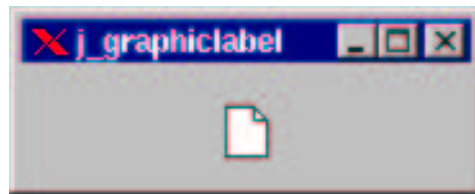
Synopsis	integer function j_graphiclabel (integer obj , character*(*) str)
Argumente	obj integer str character*(*)
Beschreibung	Die Funktion erzeugt ein neues Label mit einem Icon als Inhalt. Die Datei, die das Icon enthält, ist im Parameter filename anzugeben. Das Icon muß im GIF oder JPEG-Format vorliegen. Die Funktion liefert die Eventnummer des Labels zurück, oder im Fehlerfall -1.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame("j_graphiclabel")
label = j_graphiclabel(frame,"new.gif")
:

```



hasfocus

Synopsis	integer function j_hasfocus (integer obj)
Argumente	obj integer
Beschreibung	Liefert <code>.true.</code> zurück, falls die Komponente den Focus besitzt, sonst <code>.false.</code> .
Zielkomponenten	Focuslistener

helpmenu

Synopsis integer function **j_helpmenu** (integer obj , character*(*) label)

Argumente obj integer
 label character*(*)

Beschreibung Die Funktion erzeugt einen Hilfe-Menueintrag mit der Beschriftung **label** in der Komponenten **obj**, und liefert dessen Eventnummer zurück. An diese Menueintrag können in weiteren Schritten MenuItem gebunden werden. Eine Menueintrag selbst erzeugt keinen Event. Hilfe Menü unterscheiden sich von normalen Menüs dadurch, daß sie je nach Oberfläche auch rechteckig erscheinen können. Zudem sind als Zielkomponenten nur Menubars zulässig.

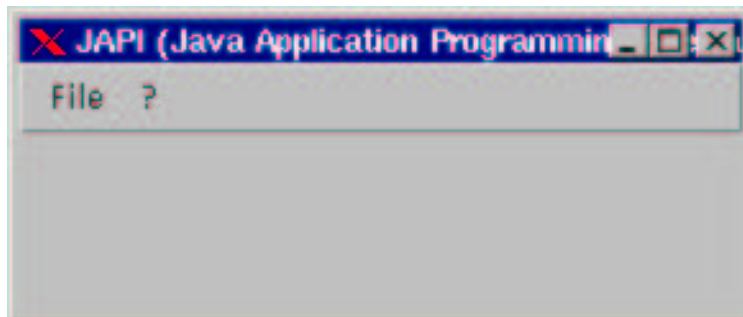
Zielkomponenten Menubar

Beispiel

```

:
frame = j_frame("Menu Komponenten")
menubar = j_menubar(frame)
file= j_menu(menubar,"File")
help= j_helpmenu(menubar,"?")
:

```



hide

Synopsis	procedure j_hide (integer obj)
Argumente	obj integer
Beschreibung	Zeigt die Komponente obj nicht mehr auf dem Bildschirm an.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

hscrollbar

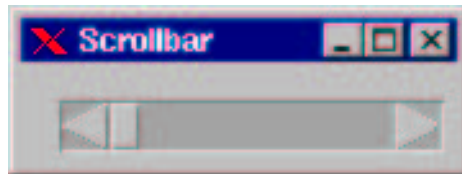
Synopsis	integer function j_hscrollbar (integer obj)
Argumente	obj integer
Beschreibung	Diese Funktion erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Ist die Zielkomponente obj vom Typ Scrollpane, so wird die Eventnummer der vorhandenen horizontalen Scrollbar zurückgegeben. Eine Scrollbar erzeugt einen Event, sobald der Schieber bewegt wird.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame, Scrollpane

Beispiel

```

:
scroll=j_hscrollbar(frame)
call j_setpos(scroll,20,40)
call j_setsize(scroll,150,20)
:

```



image

Synopsis integer function **j_image** (integer width , integer height)

Argumente width integer
 height integer

Beschreibung Diese Funktion erzeugt einen neues (Memory) Image mit der Breite **width** Pixel und einer Höhe von **height** Pixel. Die Funktion liefert die Eventnummer des Image zurück, oder im Fehlerfall -1 . Ein Image erzeugt keinen Event

Beispiel

```
:  
image = j_image(200,200)  
:
```

insert

Synopsis	integer function j_insert (integer obj , integer pos , character*(*) label)
Argumente	obj integer pos integer label character*(*)
Beschreibung	fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
Zielkomponenten	List, Choice

inserttext

Synopsis	procedure j_inserttext (integer obj , character*(*) text , integer pos)
Argumente	obj integer text character*(*) pos integer
Beschreibung	Fügt zusätzlichen Text text an der Position pos ein.
Zielkomponenten	Textarea

isparent

Synopsis	integer function j_isparent (integer obj , integer cont)
Argumente	obj integer cont integer
Beschreibung	Liefert <code>.true.</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>.false.</code>
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

isresizable

Synopsis	integer function j_isresizable (integer obj)
Argumente	obj integer
Beschreibung	Liefert <code>.false.</code> zurueck, falls die Komponente eine fixe Größe besitzt, <code>.true.</code> sonst.
Zielkomponenten	Dialog, Frame

isselect

Synopsis	integer function j_isselect (integer obj , integer item)
Argumente	obj integer item integer
Beschreibung	Liefert <code>.true.</code> zurück, falls das Item mit dem Index item selectiert ist, sonst <code>.false.</code>
Zielkomponenten	List

isvisible

Synopsis	integer function j_isvisible (integer obj)
Argumente	obj integer
Beschreibung	Liefert <code>.true.</code> zurück, falls obj sichtbar ist, sonst <code>.false.</code>
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

keylistener

Synopsis	integer function j_keylistener (integer obj)
Argumente	obj integer
Beschreibung	Bindet an die Komponente obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

label

Synopsis integer function **j_label** (integer obj , character*(*) label)

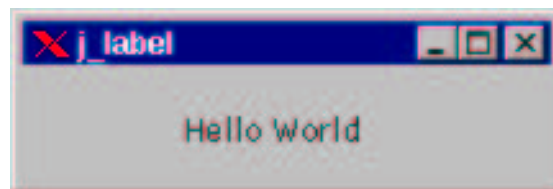
Argumente obj integer
 label character*(*)

Beschreibung Die Funktion erzeugt ein neues Label Element mit dem Textinhalt **label**. Ein Label stellt einen einzeiligen Text beliebiger Länge dar. Die Funktion liefert die Eventnummer des Labels zurück, oder im Fehlerfall -1 .

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
frame = j_frame("j_label")  
label = j_label(frame,"Hello World")  
:
```



led

Synopsis integer function **j_led** (integer obj , integer style , integer color)

Argumente obj integer
 style integer
 color integer

Beschreibung Die Funktion erzeugt ein neues LED-Element und liefert die Eventnummer der Komponente zurück. Die LED kann eine runde Form besitzen **style = J_ROUND** oder eine eckige Form besitzen **style=J_RECT**. Die Farbe der LED kann eine der vordefinierten Farben sein (zB. J_RED, J_GREEN).

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
led1 = j_led(frame,J_ROUND,J_RED)
led2 = j_led(frame,J_RECT,J_BLUE)
:

```



line

Synopsis integer function **j_line** (integer obj , integer orient , integer style , integer length)

Argumente obj integer
 orient integer
 style integer
 length integer

Beschreibung Die Funktion erzeugt ein Linienelement und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Das Element wird der Komponente zugeordnet, die durch **obj** referenziert wird. Die initiale Laenge der Linie wird durch den Parameter **length** bestimmt. Der Ruler dient der optischen Trennung anderer Komponenten.

Der Parameter **orient** gibt an, ob eine senkrechte oder waagerechte Linie gezogen werden soll:

- J_HORIZONTAL : waagerechte Linie
- J_VERTICAL : senkrechte Linie

Der Parameter **style** gibt den Linientyp vor:

- J_LINEDOWN : Der Linie scheint optisch abgesenkt.
- J_LINEUP : Der Linie scheint optisch hervorzutreten.

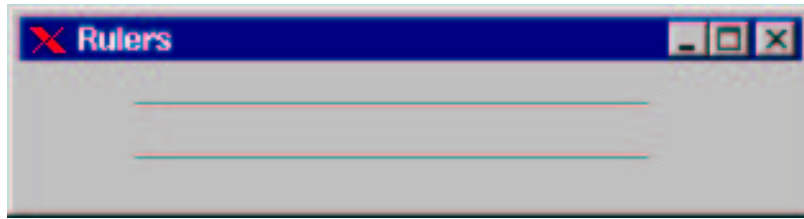
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
line1 = j_line(frame,J_HORIZONTAL,J_LINEDOWN,200)
line2 = j_line(frame,J_HORIZONTAL,J_LINEUP,200)
:

```



list

Synopsis integer function **j_list** (integer obj , integer rows)

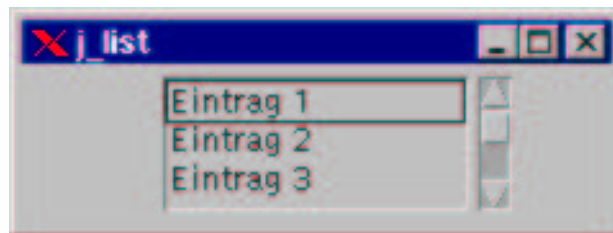
Argumente obj integer
 rows integer

Beschreibung Diese Funktion erzeugt eine neue List und gibt deren Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Eine List bietet dem Benutzer die Auswahl zwischen mehreren vordefinierten Einträgen. Es sind immer **rows** Einträge sichtbar, die anderen Einträge sind über Rollbalken zu erreichen. Eine List bietet die Möglichkeit, sowohl Einfach- als auch Mehrfachselektionen zuzulassen (siehe *j_multiplemode()*). Bei Einfachselektion liefert die List bei einem Doppelklick auf einen Eintrag einen Event. Ist eine Mehrfachselektion eingestellt, löst jeder einfache Klick auf einen Eintrag einen Event aus.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:
list = j_list(frame,3)
call j_additem(list,"Eintrag 1")
call j_additem(list,"Eintrag 2")
:
```



loadimage

Synopsis	integer function j_loadimage (character*(*) filename)
Argumente	filename character*(*)
Beschreibung	Liest das Image aus dem File filename ein und liefert dessen Eventnummer zurück. Es werden folgende Fileformate akzeptiert: <ul style="list-style-type: none">• GIF• JPEG• BMP• PPM
Beispiel	<pre>: image = j_loadimage("mandel.jpg") :</pre>

menubar

Synopsis integer function **j_menubar** (integer obj)

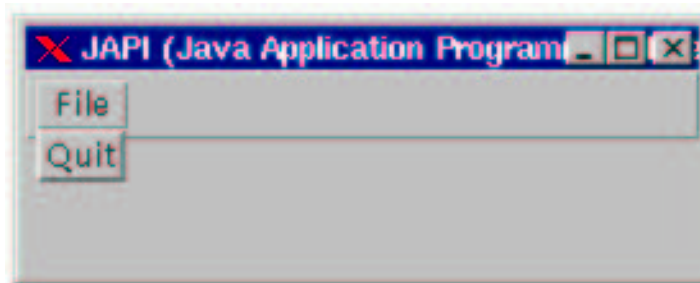
Argumente obj integer

Beschreibung Die Funktion erzeugt eine neue Menubar in der Komponente, die durch **obj** referenziert wird, und liefert dessen Eventnummer zurück. An die Menubar können in weiteren Schritten Menu Komponenten gebunden werden. Eine Menubar selbst erzeugt keinen Event. Als Zielkomponenten sind nur Frames zulässig.

Zielkomponenten Frame

Beispiel

```
:
frame  = j_frame("Menu Komponenten")
menubar = j_menubar(frame)
file   = j_menu(menubar,"File")
quit   = j_menuitem(file,"Quit")
:
```



menuitem

Synopsis integer function **j_menuitem** (integer obj , character*(*) label
)

Argumente obj integer
 label character*(*)

Beschreibung Die Funktion erzeugt ein MenuItem mit der Beschriftung **label** in der Komponente **obj**, und liefert dessen Eventnummer zurück. Eine MenuItem erzeugt einen Event, wenn es durch einen Mausklick angewählt wurde. Als Zielkomponenten sind nur Menu, Popupmenu und Helpmenu zulässig.

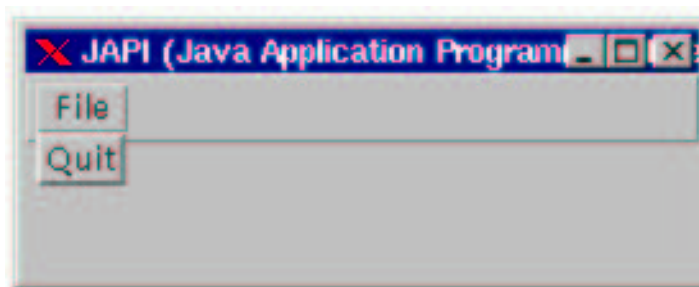
Zielkomponenten Menu, Popupmenu, Helpmenu

Beispiel

```

:
frame = j_frame("Menu Komponenten")
menubar = j_menubar(frame)
file = j_menu(menubar,"File")
quit = j_menuitem(file,"Quit")
:

```



menu

Synopsis integer function **j_menu** (integer obj , character*(*) str)

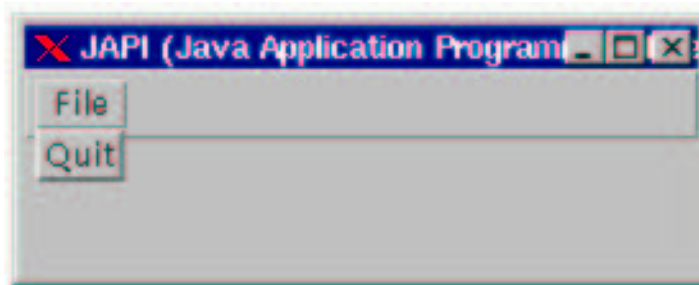
Argumente obj integer
 str character*(*)

Beschreibung Die Funktion erzeugt einen Menueintrag mit der Beschriftung **label** in der Komponente, die durch **obj** referenziert wird, und liefert dessen Eventnummer zurück. An diese Menukomponente können in weiteren Schritten MenuItem gebunden werden. Eine Menukomponente selbst erzeugt keinen Event. Als Zielkomponenten sind nur Menubars und andere Menukomponenten zulässig.

Zielkomponenten Menubar, Menu

Beispiel

```
:  
frame  = j_frame("Menu Komponenten")  
menubar = j_menubar(frame)  
file   = j_menu(menubar, "File")  
quit   = j_menuitem(file, "Quit")  
:
```



messagebox

Synopsis `procedure j_messagebox (integer obj , character*(*) title ,
character*(*) text)`

Argumente `obj integer
title character*(*)
text character*(*)`

Beschreibung Öffnet eine MessageBox mit der Überschrift **title** und dem Textinhalt **text** und liefert dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgeliefert. Eine MessageBox erzeugt einen Event, wenn das Close Icon im Fensterrahmen angeklickt wird.

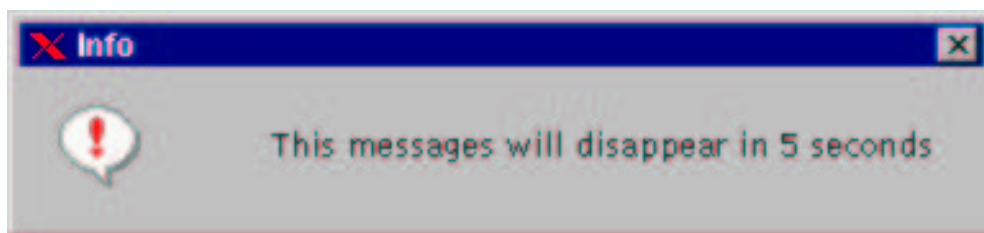
Zielkomponenten Frame

Beispiel

```

:
mbox = j_messagebox(frame,"Info","This messages will disappear in 5 seconds")
call j_sleep(5000)
call j_dispose(mbox)
:

```



meter

Synopsis integer function **j_meter** (integer obj , character*(*) title)

Argumente obj integer
 title character*(*)

Beschreibung Die Funktion erzeugt ein analoges Anzeigeeinstrument und liefert die Eventnummer der Komponente zurück. Das Instrument trägt die Beschriftung **title**. Das Instrument besitzt einen vordefinierten Bereich von 0 bis 100. Die Werte können mit `j_setmin()` und `j_setmax()` verändert werden. Ein Gefahr-Bereich ist ab dem Wert 80 voreingestellt. Dies kann mit `j_setdanger()` verändert werden.

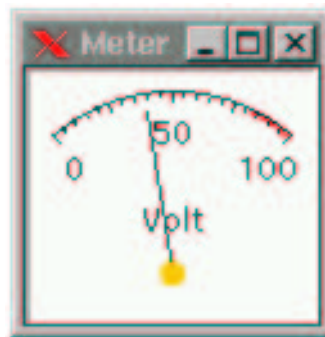
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
meter = j_meter(frame,"Volt")
call j_setvalue(meter,40)
:

```



mouselistener

Synopsis	integer function j_mouselistener (integer obj , integer kind)
Argumente	obj integer kind integer
Beschreibung	<p>Bindet an die Komponente obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für kind sind:</p> <ul style="list-style-type: none"> • J_ENTERED : Der Mouselistener liefert einen Event, wenn der Mauszeiger die Komponente obj betritt. • J_MOVED : Der Mouselistener liefert einen Event, wenn der Mauszeiger sich innerhalb der Komponenten obj bewegt. • J_EXITED : Der Mouselistener liefert einen Event, wenn der Mauszeiger die Komponente obj verläßt. • J_PRESSED : Der Mouselistener liefert beim Drücken einer Mousetaste einen Event. • J_DRAGGED : Der Mouselistener liefert einen Event, wenn der Mauszeiger sich innerhalb der Komponenten obj mit gedrückter Taste bewegt. • J_RELEASED : Der Mouselistener liefert beim Loslassen einer Mousetaste einen Event. • J_DOUBLECLICK : Der Mouselistener liefert bei einem Doppelclick einen Event.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

multiplemode

Synopsis	integer function j_multiplemode (integer obj , integer bool)
Argumente	obj integer bool integer
Beschreibung	die Komponente erlaubt mehrfache Selectierung, falls bool .true. ist.
Zielkomponenten	List

nextaction

Synopsis integer function **j_nextaction** ()

Beschreibung Wartet auf den nächsten Benutzerevent und liefert diesen zurück.

pack

Synopsis procedure **j_pack** (integer obj)

Argumente obj integer

Beschreibung Diese Prozedur veranlaßt den Layoutmanager von **obj** die minimale Größe zu berechnen. Ist die Komponente sichtbar, so wird das neue Layout direkt angezeigt. Der Layoutmanager erfragt von allen enthaltenen Komponenten deren initiale bzw. explizit zugewiesene Größe, und berechnet daraus ein neues Layout.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
call j_setflowlayout(jframe,J_HORIZOMTAL)  
canvas = j_canvas(frame,200,50)  
call j_setnamedcolorbg(canvas,J_RED)  
call j_pack(frame)  
:
```



panel

Synopsis integer function **j_panel** (integer obj)

Argumente obj integer

Beschreibung Diese Funktion erzeugt ein neues panel und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Das Panel wird der Komponente zugeordnet, die durch **obj** referenziert wird. Ein Panel ist ein (unsichtbares) Behälterelement, in dem andere Elemente angeordnet werden können. Da einem Panel auch ein Layoutmanager zugewiesen werden kann, können mit Hilfe der Panels komplexe hirarchisch aufgebaute Benutzerschnittstellen erstellt werden. Ein Panel liefert bei einer Größenänderung einen Event.

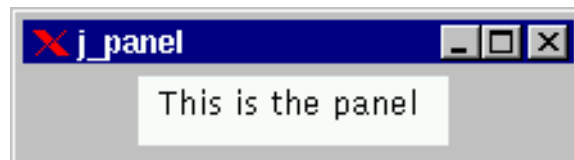
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
panel = j_panel(frame)
call j_setnamedcolorbg(panel,J_WHITE)
call j_setpos(panel,50,30)
label = j_label(panel,"This is the panel")
call j_setpos(label,0,0)
:

```



popupmenu

Synopsis integer function **j_popupmenu** (integer obj , character*(*)
label)

Argumente obj integer
label character*(*)

Beschreibung Diese Funktion erzeugt ein neues Popupmenu und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Ein Popupmenu ist funktional idemisch mit einem Menu, es ist allerdings an keine Menuleiste gebunden. Daher kann ein Popupmenu prinzipiell an jeder Position innerhalb der Oberfläche platziert werden. Wie an Menus kann auch an ein Popupmenu MenuItem's gebunden werden, die bei Anklicken des Popupmenus sichtbar werden.

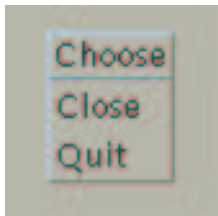
Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

Beispiel

```

:
choose = j_popupmenu(frame,"Choose")
close  = j_menuitem(choose,"Close")
quit   = j_menuitem(choose,"Quit")
call j_showpopup(choose,100,100)
:

```



printer

Synopsis integer function **j_printer** (integer frame)

Argumente frame integer

Beschreibung Diese Funktion erzeugt ein neues Objekt, das eine Papierseite eines Druckes repräsentiert. Bis auf wenige Ausnahmen ist ein Printer Objekt funktional einem Canvas gleichgestellt. Die Funktion liefert die Eventnummer des Printers zurück, oder im Fehlerfall -1. Ein Printer erzeugt keinen Event.

Zielkomponenten Frame

Beispiel

```
:
printer = j_printer(frame)
call j_drawimage(printer,image,100,100)
:
```

print

Synopsis	procedure j_print (integer obj)
Argumente	obj integer
Beschreibung	<p>Die Prozedur druckt den Inhalt der Komponente obj. Ist die Komponente von Typ Container, so wird der gesamte Inhalt des Containers gedruckt. Nach dem Aufruf dieser Funktion erscheint ein Plattformabhängiger Print Dialog, in dem weitere Optionen (beispielsweise das Speichern in eine Datei) bereitgestellt werden. Unter X-Windows besitzt der Ausdruck immer Motif Look, unabhängig von der Bildschirmdarstellung.</p> <p>Ist die Komponente von Type j_printer, dann wird die aktuelle Seite geschlossen, und eine neue Seite begonnen. Die Seiten werden noch nicht auf dem Drucker ausgegeben. Um den Druck zu starten muss j_dispose(printer) aufgerufen werden.</p>
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Canvas, Image, Printer
Beispiel	<pre> : frame = j_frame("j_textfield") text = j_textfield(frame,30) : call j_print(frame) : </pre>



progressbar

Synopsis integer function **j_progressbar** (integer obj , integer orient)

Argumente obj integer
 orient integer

Beschreibung Die Funktion erzeugt eine Progressbar und liefert die Eventnummer der Komponente zurück. Die Ausrichtung kann J_HORIZONTAL oder J_VERTICAL sein. Das Instrument besitzt einen vordefinierten Bereich von 0 bis 100. Die Werte können mit *j_setmin()* und *j_setmax()* verändert werden.

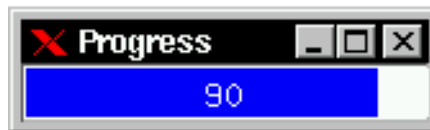
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
progress = j_progressbar(frame,J_HORIZONTAL)
call j_setvalue(progress,90)
:

```



quit

Synopsis procedure **j_quit** ()

Beschreibung Beendet die Verbindung zum JAPI Kernel.

radiobutton

Synopsis integer function **j_radiobutton** (integer obj , character*(*)
label)

Argumente obj integer
 label character*(*)

Beschreibung Diese Funktion erzeugt einen Radiobutton mit dem Textinhalt **label** und liefert dessen Eventnummer zuruck. Im Fehlerfall wird -1 zurückgegeben. Ein Radiobutton muss immer in einer Radiogroup eingebettet sein. Innerhalb dieser Radiogroup kann von allen enthaltenen Radiobuttons immer nur maximal ein Button aktiviert sein. Ein Radiobutton liefert immer dann einen Event, wenn sich sein Zustand ändert.

Zielkomponenten Radiogroup

Beispiel

```

:
radiogroup = j_radiogroup(frame)
radio1     = j_radiobutton(radiogroup,"Radiobutton 1")
radio2     = j_radiobutton(radiogroup,"Radiobutton 2")
:

```



radiogroup

Synopsis	integer function j_radiogroup (integer obj)
Argumente	obj integer
Beschreibung	Diese Funktion erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück. Eine Radiogroup ist eine virtuelle Komponente, die dazu dient, Radiobutton funktionell zusammenzuschalten. Von allen Radiobutton, die in einer Radiogroup zusammengefasst sind, kann immer nur maximal ein Button aktiviert sein.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
radiogroup = j_radiogroup(frame)  
radio1     = j_radiobutton(radiogroup,"Radiobutton 1")  
radio2     = j_radiobutton(radiogroup,"Radiobutton 2")  
:
```



random

Synopsis integer function **j_random** ()

Beschreibung Liefert eine Pseudo Zufallszahl zwischen 0 und 2147483647 ($2^{31} - 1$) zurück .

releaseall

Synopsis	procedure j_releaseall (integer obj)
Argumente	obj integer
Beschreibung	Gibt alle Elemente aus der Komponenten obj frei.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

release

Synopsis	procedure j_release (integer obj)
Argumente	obj integer
Beschreibung	Gibt die Komponente obj aus dem umgebenden Container Element frei.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

removeall

Synopsis integer function **j_removeall** (integer obj)

Argumente obj integer

Beschreibung entfernt alle Items aus der Komponenten .

Zielkomponenten List, Choice

removeitem

Synopsis	integer function j_removeitem (integer obj , character*(*) item)
Argumente	obj integer item character*(*)
Beschreibung	entfernt den ersten Eintrag item aus der Komponenten .
Zielkomponenten	List, Choice

remove

Synopsis integer function **j_remove** (integer obj , integer item)

Argumente obj integer
 item integer

Beschreibung entfernt das Item mit dem Index **item** aus der Komponenten .

Zielkomponenten List, Choice

replacetext

Synopsis	procedure j_replacetext (integer obj , character*(*) text , integer start , integer end)
Argumente	obj integer text character*(*) start integer end integer
Beschreibung	Ersetzt den Text zwischen Position start bis end durch text .
Zielkomponenten	Textarea

saveimage

Synopsis integer function **j_saveimage** (integer obj , character*(*) filename , integer filetype)

Argumente obj integer
 filename character*(*)
 filetype integer

Beschreibung Speichert den Inhalt der Komponenten in einem File mit dem Namen **filename**. Es werden zwei Fileformate unterstützt:

- BMP
- PPM

Beispiel

```
:  
if(j_saveimage(canvas,"mandel.bmp",J_BMP) .eq. false)  
  write (*,*) "Error saving Bitmap file"  
:
```


scrollpane

Synopsis integer function **j_scrollpane** (integer obj)

Argumente obj integer

Beschreibung Diese Funktion erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Eine Scrollpane ist eine Behälterkomponente, die ein weiteres Objekt aufnehmen kann. Dabei kann das aufgenommene Objekt größere Ausmaße haben als die Scrollpane. Über zwei automatisch angepaßte Scrollbars, kann das enthaltene Object gescrollt werden. Ein Layoutmanager kann nicht in eine Scrollpane integriert werden. Soll mehr als eine Komponente aufgenommen werden, so ist in die Scrollpane zunächst ein Panel zu integrieren. Eine Scrollpane liefert bei einer Größenänderung einen Event.

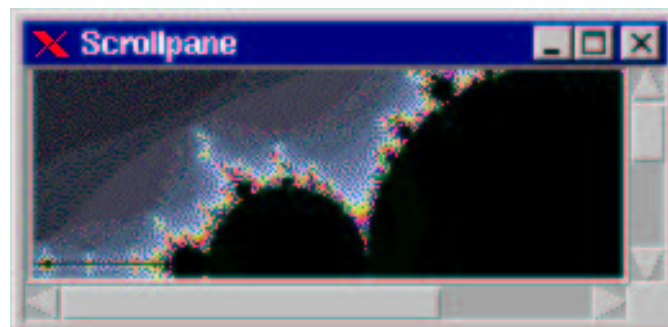
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
scrollpane = j_scrollpane(frame)
image = j_graphiclabel(scrollpane,"mandel.gif")
call j_setsize(scrollpane,240,100)
:

```



selectall

Synopsis	procedure j_selectall (integer obj)
Argumente	obj integer
Beschreibung	Selectiert den gesamten Textinhalt der Komponenten .
Zielkomponenten	Textarea, Textfield

select

Synopsis integer function **j_select** (integer obj , integer item)

Argumente obj integer
 item integer

Beschreibung selectiert das Item mit dem Index **item**.

Zielkomponenten List, Choice

selecttext

Synopsis	procedure j_selecttext (integer obj , integer start , integer end)
Argumente	obj integer start integer end integer
Beschreibung	Selectiert den Text von Position start bis end .
Zielkomponenten	Textarea, Textfield

seperator

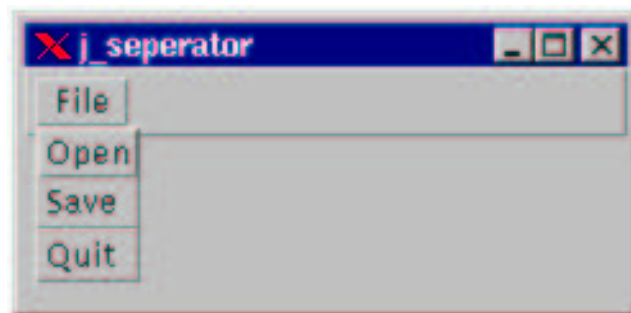
Synopsis	procedure j_seperator (integer obj)
Argumente	obj integer
Beschreibung	Diese Prozedur erzeugt eine Trennlinie in einem Menu, das durch obj referenziert wird. Trennlinien dienen ausschließlich der optischen Trennung von MenuItemen, und besitzen keine weitere Funktionalität.
Zielkomponenten	Menu, HelpMenu, Popupmenu

Beispiel

```

:
file = j_menu(menubar,"File")
open = j_menuitem(file,"Open")
save = j_menuitem(file,"Save")
call j_seperator(file)
quit = j_menuitem(file,"Quit")
:

```



setalign

Synopsis	procedure j_setalign (integer obj , integer align)
Argumente	obj integer align integer
Beschreibung	Setzt die Ausrichtung innerhalb der Komponenten obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setblockinc

Synopsis	integer function j_setblockinc (integer obj , integer val)
Argumente	obj integer val integer
Beschreibung	Stellt die Blockänderung von der Komponenten auf den Wert von val ein.
Zielkomponenten	Scrollbar

setborderlayout

Synopsis	procedure j_setborderlayout (integer obj)
Argumente	obj integer
Beschreibung	Integriert einen BorderLayout Manager in die Komponente obj .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setborderpos

Synopsis	procedure j_setborderpos (integer obj , integer pos)
Argumente	obj integer pos integer
Beschreibung	Positioniert die Komponente obj an die angegebene Position pos . Im umgebenden Container muß ein BorderLayout Manager eingestellt sein.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setcolorbg

Synopsis procedure **j_setcolorbg** (integer obj , integer r , integer g , ,
integer b)


Argumente obj integer
 r integer
 g, integer
 b integer

Beschreibung Diese Prozedur setzt in dem Komponenten **obj** die Hintergrundfarbe. Als Parameter sind der **rot**, **grün** und **blau**-Anteil der Farbe zu übergeben. Unabhängig von der tatsächlichen vorhandenen Farbtiefe des Systems wird immer ein 24 Bit Farbwert angenommen. Dieser teilt sich in je 8 Bit für den Rot, den Grün und den Blau-Kanal auf. Somit können diese Parameter Werte zwischen 0 (dunkel) und 255 (hell) annehmen. Das Setzen einer neuen Hintergrundfarbe in einem Canvas bewirkt, das der Inhalt des Cavas überzeichnet wird.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```
:
button = j_button(frame,"Hello World")
call j_setcolorbg(button,150,0,0)
call j_settext(button,"Hello World")
:
```



setcolor

Synopsis procedure **j_setcolor** (integer obj , integer r , integer g , integer b)

Argumente

obj	integer
r	integer
g,	integer
b	integer

Beschreibung Diese Prozedur setzt in der, durch **obj** referenzierten, Komponente die Vordergrundfarbe. Als Parameter sind der **rot**, **grün** und **blau**-Anteil der Farbe zu übergeben. Unabhängig von der tatsächlichen vorhandenen Farbtiefe des Systems wird immer ein 24 Bit Farbwert angenommen. Dieser teilt sich in je 8 Bit für den Rot, den Grün und den Blau-Kanal auf. Somit können diese Parameter Werte zwischen 0 (dunkel) und 255 (hell) annehmen. Die Vordergrundfarbe der Komponenten bleibt solange aktiv, bis diese durch einen neuen Aufruf einer Farbfunktion überschrieben wird.


Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```

:
button = j_button(frame,"Hello World")
call j_setcolor(button,150,0,0)
call j_settext(button,"Hello World")
:

```



Hello World

setcolumns

Synopsis procedure **j_setcolumns** (integer obj , integer columns)

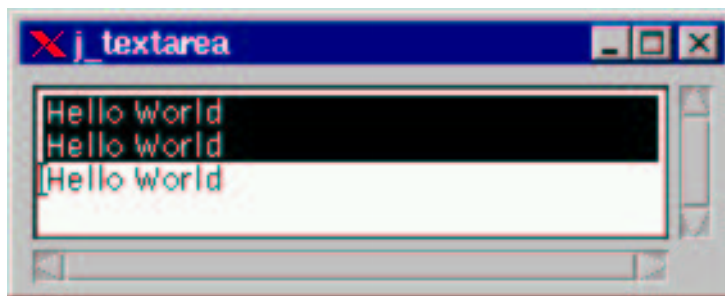
Argumente obj integer
 columns integer

Beschreibung Setzt die Anzahl Spalten von **obj** auf **columns**.

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

```
:  
text = j_text(frame,10,4)  
call j_setcolumns(text,30)  
:
```



setcurpos

Synopsis procedure **j_setcurpos** (integer obj , integer pos)

Argumente obj integer
 pos integer

Beschreibung Setzt den Text Cursor auf die Position **pos**.

Zielkomponenten Textarea, Textfield

setcursor

Synopsis integer function **j_setcursor** (integer obj , integer cursor)

Argumente obj integer
 cursor integer

Beschreibung Setzt den Cursor der Komponenten **obj** auf **cursor**.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice,
Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window,
Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar,
Meter, Sevenssegment

setdanger

Synopsis procedure **j_setdanger** (integer obj , integer val)

Argumente obj integer
 val integer

Beschreibung Stellt den Warnlevel der Komponenten **obj** auf **val**.

Zielkomponenten Meter

setdebug

Synopsis	procedure j_setdebug (integer level)
Argumente	level integer
Beschreibung	<p>Diese Prozedur setzt den Debuglevel auf level. Ab einem Level größer 0 erscheint ein Debugwindow, in dem die JAPI Calls protokolliert werden. Über den Parameter level kann die Protokolltiefe eingestellt werden. j_setdebug ist die einzige Funktion, die aufgerufen werden kann, bevor j_start() aufgerufen wurde. Die Levelbenen im einzelnen:</p> <ul style="list-style-type: none">0: keine Ausgabe (default Wert)1: Rückmeldung der konstruktiven Funktionen. Nur das Erzeugen der graphischen Objekte wird protokolliert.2: Wie 1, zusätzliche Ausgabe aller Aktionen, die vom Benutzer ausgeführt werden.3: Wie 2, zusätzlich werden alle weiteren Funktionen (außer den graphischen Befehlen) protokolliert.4: Wie 3, zusätzlich mit allen graphischen Befehlen.

setechochar

Synopsis	procedure j_setechochar (integer obj , character chr)
Argumente	obj integer chr character
Beschreibung	Bestimmt das Zeichen chr als Echo Zeichen für alle Benutzereingaben in die Komponente .
Zielkomponenten	Textfield

seteditable

Synopsis	procedure j_seteditable (integer obj , integer bool)
Argumente	obj integer bool integer
Beschreibung	Ist bool <code>.true.</code> , so ist die Komponente editierbar, ansonsten ist die Komponente nur lesbar für den Benutzer.
Zielkomponenten	Textarea, Textfield

setfixlayout

Synopsis	procedure j_setfixlayout (integer obj)
Argumente	obj integer
Beschreibung	Integriert einen Fixlayout Manager in die Komponente obj (voreingestellte Layoutmanager).
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setflowfill

Synopsis	procedure j_setflowfill (integer obj , integer bool)
Argumente	obj integer bool integer
Beschreibung	Setzt die Höhe (Breite) aller enthaltenen Komponenten auf die Höhe (Breite) von der Komponenten obj . Ein Flowlayout Manager wird vorausgesetzt.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setflowlayout

Synopsis	procedure j_setflowlayout (integer obj , integer align)
Argumente	obj integer align integer
Beschreibung	Integriert einen Flowlayout Manager in die Komponente obj mit der Ausrichtung align .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setfocus

Synopsis	integer function j_setfocus (integer obj)
Argumente	obj integer
Beschreibung	Weist den Input Focus der Komponenten obj zu.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setfontname

Synopsis	procedure j_setfontname (integer obj , integer name)
Argumente	obj integer name integer
Beschreibung	Diese Prozedur setzt den Fonttyp in der Komponenten obj , auf name . Der Parameter name kann folgende Werte annehmen: <ul style="list-style-type: none"> • J_COURIER : Courier (Monospaced) • J_HELVETIA : Helvetia (Sans Serif) • J_TIMES : Times (Serif) • J_DIALOGIN : Dialog In • J_DIALOGOUT : Dialog Out
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu

Beispiel

```

:
label = j_label(jframe,"Hello World")
call j_setfontname(label,J_HELVETIA)
:

```



setfontsize

Synopsis	procedure j_setfontsize (integer obj , integer size)
Argumente	obj integer size integer
Beschreibung	Diese Prozedur setzt die Fontgröße in der Komponenten, obj , auf height . Die Größe size , kann prinzipiell beliebige positive Zahlen annehmen. Die Größe des Fonts hat die Einheit Pixel.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu
Beispiel	<pre>: label = j_label(jframe,"Hello World") call j_setfontsize(label,24) :</pre>



setfontstyle

Synopsis	procedure j_setfontstyle (integer obj , integer style)
Argumente	obj integer style integer
Beschreibung	Diese Prozedur setzt den Fontstyle in der Komponenten, die durch obj referenziert wird, auf style . Für den Parameter style sind folgende Werte möglich: <ul style="list-style-type: none"> • J_PLAIN : normal • J_BOLD : fett • J_ITALIC : italic • J_BOLD + J_ITALIC : italic fett
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu
Beispiel	<pre> : label = j_label(jframe,"Hello World") call j_setfontstyle(label,J_BOLD+J_ITALIC) : </pre>



setfont

Synopsis	procedure j_setfont (integer obj , integer name , integer style , integer size)								
Argumente	<table> <tr> <td>obj</td> <td>integer</td> </tr> <tr> <td>name</td> <td>integer</td> </tr> <tr> <td>style</td> <td>integer</td> </tr> <tr> <td>size</td> <td>integer</td> </tr> </table>	obj	integer	name	integer	style	integer	size	integer
obj	integer								
name	integer								
style	integer								
size	integer								
Beschreibung	<p>Diese Prozedur setzt erzeugt einen neuen Font in der Komponenten, die durch obj referenziert wird. Der Font wird durch die folgenden drei Parameter spezifiziert:</p> <p>Der Parameter name kann folgende Werte annehmen:</p> <ul style="list-style-type: none"> • J_COURIER : Courier (Monospaced) • J_HELVETIA : Helvetia (Sans Serif) • J_TIMES : Times (Serif) • J_DIALOGIN : Dialog In • J_DIALOGOUT : Dialog Out <p>Für den Parameter style sind folgende Werte möglich:</p> <ul style="list-style-type: none"> • J_PLAIN : normal • J_BOLD : fett • J_ITALIC : italic • J_BOLD + J_ITALIC : italic fett <p>Die Größe size wird als Integerzahl übergeben. Sie kann prinzipiell beliebige positive Zahlen annehmen.</p>								
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu								

Beispiel

```
:  
label = j_label(jframe,"Hello World")  
call j_setfont(label,J_TIMES,J_PLAIN,18)  
:
```



setgridlayout

Synopsis	procedure j_setgridlayout (integer obj , integer row , integer col)
Argumente	obj integer row integer col integer
Beschreibung	Integriert einen Gridlayout Manager in die Komponente obj mit row Reihen und col Spalten.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

sethgap

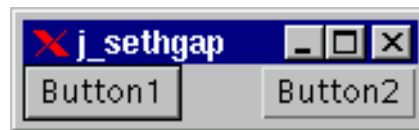
Synopsis	procedure j_sethgap (integer obj , integer hgap)
Argumente	obj integer hgap integer
Beschreibung	Setzt den horizontalen Abstand innerhalb der, durch obj referenzierten, Komponenten auf hgap Pixel. Die Komponente muss vom Typ Container sein (siehe Zielkomponenten). Weiterhin sollte in der Komponente ein Layoutmanager definiert sein, da diese Funktion nur in Verbindung mit einem Layoutmanager Sinn macht. Der Layoutmanager ordnet nun alle Objekte so an, daß deren horizontaler Abstand zueinander genau hgap Pixel beträgt.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
call j_flowlayout(frame,J_HORIZONTAL)
button1 = j_button(frame,"Button1")
button2 = j_button(frame,"Button2")
call j_sethgap(frame,30)
:

```



seticon

Synopsis procedure **j_seticon** (integer frame , integer icon)

Argumente frame integer
 icon integer

Beschreibung Ordnet dem **frame** das Icon **icon** zu. Eine iconisierte Darstellung wird nicht von allen Plattformen unterstützt.

Zielkomponenten Frame

Beispiel

```
:  
frame = j_frame("Hello World")  
call j_seticon(frame,j_loadimage("icon.gif"))  
:
```

setimage

Synopsis procedure **j_setimage** (integer obj , integer image)

Argumente obj integer
 image integer

Beschreibung Ordnet dem **obj** das Bild **image** zu.

Zielkomponenten Graphicbutton, Graphiclabel

Beispiel

```
:  
label = j_graphiclabel(frame,"mandel.gif")  
image = j_image("new.gif")  
call j_setimage(label,image)  
:
```



setinsets

Synopsis procedure **j_setinsets** (integer obj , integer top , integer bottom
 , integer left , integer right)

Argumente	obj	integer
	top	integer
	bottom	integer
	left	integer
	right	integer

Beschreibung Setzt den inneren Rahmen auf die angegebenen Größen.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame("j_getinsets")
call j_setinsets(frame,30,10,10,10)
:
> 25 5 5 6
    
```



setmax

Synopsis	integer function j_setmax (integer obj , integer val)
Argumente	obj integer val integer
Beschreibung	Stellt die obere Grenze der Komponenten auf den Wert von val ein.
Zielkomponenten	Scrollbar, Meter, Progress

setmin

Synopsis	integer function j_setmin (integer obj , integer val)
Argumente	obj integer val integer
Beschreibung	Stellt die untere Grenze der Komponenten auf den Wert von val ein.
Zielkomponenten	Scrollbar, Meter, Progress

setnamedcolorbg

Synopsis	procedure j_setnamedcolorbg (integer obj , integer color)
Argumente	obj integer color integer
Beschreibung	Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

setnamedcolor

Synopsis	procedure j_setnamedcolor (integer obj , integer color)
Argumente	obj integer color integer
Beschreibung	Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

setnolayout

Synopsis	procedure j_setnolayout (integer obj)
Argumente	obj integer
Beschreibung	Entfernt den aktuellen Layout Manager aus der Komponenten obj .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

start

Synopsis procedure **j_start** (integer port)

Argumente port integer

Beschreibung Diese Funktions ersetzt den default Port durch den übergebenen Parameter. Damit kann mehr als ein Kernel auf einem Rechner gestartet werden. Diese Funktion muss vor j_start() aufgerufen werden.

Beispiel

```
:
call j_setport(12345)
if( .not. j_start() ) then
:
```

setpos

Synopsis	procedure j_setpos (integer obj , integer xpos , integer ypos)
Argumente	obj integer xpos integer ypos integer
Beschreibung	Verschiebt die Komponente obj zur angegebenen Koordinate (xpos,ypos) an.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setradiogroup

Synopsis	integer function j_setradiogroup (integer rbutton, , integer rgroup)
Argumente	rbutton, integer rgroup integer
Beschreibung	Weist dem Radiobutton rbutton die Radiogroup rgroup zu. Der Radiobutton wird aus der alten Radiogroup entfernt.
Zielkomponenten	Radiobutton

setresizable

Synopsis	procedure j_setresizable (integer obj , integer resizable)
Argumente	obj integer resizable integer
Beschreibung	Ist resizable <code>.false.</code> , kann die Größe der Komponenten nicht vom Benutzer verändert werden.
Zielkomponenten	Dialog, Frame
Beispiel	<pre>: frame = j_frame("fixsized Frame") call j_setrezisable(frame,.false.) :</pre>

setrows

Synopsis procedure **j_setrows** (integer obj , integer rows)

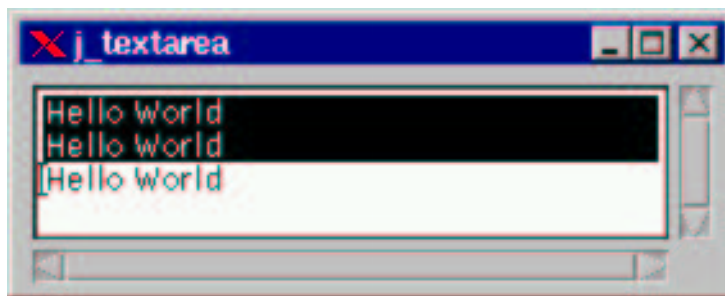
Argumente obj integer
 rows integer

Beschreibung Setzt die Anzahl Reihen von **obj** auf **rows**.

Zielkomponenten Textarea, Gridlayout

Beispiel

```
:  
text = j_text(frame,30,10)  
call j_setrows(text,4)  
:
```



setshortcut

Synopsis	procedure j_setshortcut (integer obj , character chr)
Argumente	obj integer chr character
Beschreibung	Bestimmt das Zeichen chr als Shortcut für die Komponente .
Zielkomponenten	MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

setsize

Synopsis procedure **j_setsize** (integer obj , integer width , integer height)

Argumente obj integer
 width integer
 height integer

Beschreibung Mit dieser Prozedur wird einer Komponente, die durch **obj** referenziert wird, eine neue Größe zugewiesen. Die Komponente erhält eine Breite von **width** Pixeln und eine Höhe von **height** Pixeln zugewiesen. Obwohl diese Prozedur prinzipiell auf alle Komponenten anwendbar ist, hat sie bei einigen Komponenten keine Auswirkung. So hat eine Choice beispielweise eine unveränderliche Höhe. Diese Höhe wird durch die Peer Komponente der jeweiligen Oberfläche vorgegeben. Eine weitere Ausnahme stellen die beiden Text Komponenten dar. Bei diesen Komponenten erfolgt die Größenangabe nicht in Bildschirmpunkten, sondern in der Anzahl der sichtbaren Textzeichen.

Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```
:
button = j_button(frame,"Button")
call j_setsize(button,100,100)
:
```



setslidesize

Synopsis integer function **j_setslidesize** (integer obj , integer val)

Argumente obj integer
 val integer

Beschreibung Setzt die Größe des Schiebers auf den Wert von **val**.

Zielkomponenten Scrollbar

setstate

Synopsis procedure **j_setstate** (integer obj , integer bool)


Argumente obj integer
 bool integer

Beschreibung Ist **bool** .true. so wird die Komponente selectiert.

Zielkomponenten Checkbox, Radiobutton, CheckmenuItem, Led

settext

Synopsis	procedure j_settext (integer obj , character*(*) str)
Argumente	obj integer str character*(*)
Beschreibung	Setzt den Textinhalt in der, durch obj referenzierten, Komponenten auf den Inhalt des Parameter str . Ist die Komponente vom Typ Frame oder Dialog, so wird der übergebene Text im Rahmen des Fensters eingetragen. Ansonsten wird der Textinhalt der Komponenten gesetzt. Der übergebene Text sollte keine Zeilenumbrüche enthalten, da fast alle Komponenten nur einzeilige Texte darstellen können. Einzige Ausnahme ist die Textarea Komponente, die mehrere Zeilen darstellen kann. Wird an diese Komponente ein Text übergeben, so darf dieser natürlich auch Zeilenumbrüche enthalten.
Zielkomponenten	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu, Textarea, Textfield
Beispiel	<pre> : button = j_button(frame,"Hello World") call j_settext(button,"Goodbye") : </pre>



setunitinc

Synopsis	integer function j_setunitinc (integer obj , integer val)
Argumente	obj integer val integer
Beschreibung	Stellt die minimale Wertänderung von die Komponente auf den Wert von val ein.
Zielkomponenten	Scrollbar

setvalue

Synopsis procedure **j_setvalue** (integer obj , integer val)

Argumente obj integer
 val integer

Beschreibung Stellt den aktuellen Wert der Komponenten **obj** auf **val**.

Zielkomponenten Scrollbar, Progress, Meter, Sevenssegment

setvgap

Synopsis procedure **j_setvgap** (integer obj , integer vgap)

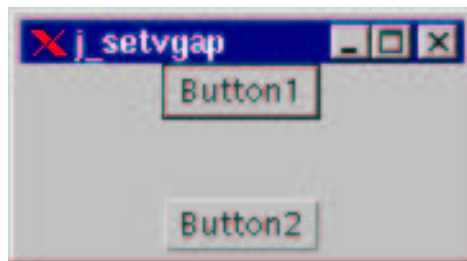
Argumente obj integer
 vgap integer

Beschreibung Setzt den vertikalen Abstand innerhalb der, durch **obj** referenzieren, Komponenten auf **vgap** Pixel. Die Komponente muss vom Typ Container sein (siehe Zielkomponenten). Weiterhin sollte in der Komponente ein Layoutmanager definiert sein, da diese Funktion nur in Verbindung mit einem Layoutmanager Sinn macht. Der Layoutmanager ordnet nun alle Objekte so an, daß deren vertikaler Abstand zueinander genau **vgap** Pixel beträgt.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
call j_setflowlayout(frame,J_VERTICAL)  
button1 = j_button(frame,"Button1")  
button2 = j_button(frame,"Button2")  
call j_setvgap(frame,30)  
:
```



setxor

Synopsis	procedure j_setxor (integer obj , integer bool)
Argumente	obj integer bool integer
Beschreibung	Schaltet den Painting Mode auf XOR, falls bool = .true. ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
Zielkomponenten	Canvas, Image, Printer

sevensegment

Synopsis integer function **j_sevensegment** (integer obj , integer color)

Argumente obj integer
 color integer

Beschreibung Die Funktion erzeugt eine Siebensegment-Anzeige und liefert die Eventnummer der Komponente zurück. Die Farbe der Siebensegment-Anzeige kann eine der vordefinierten Farben sein (zB. J_RED, J_GREEN).

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```
:  
seven = j_sevensegment(frame,J_GREEN)  
call j_setvalue(seven,5)  
:
```



showpopup

Synopsis	procedure j_showpopup (integer obj , integer xpos , integer ypos)
Argumente	obj integer xpos integer ypos integer
Beschreibung	Zeigt die Komponente an der Koordinate (xpos,ypos) an.
Zielkomponenten	Popupmenu

show

Synopsis	procedure j_show (integer obj)
Argumente	obj integer
Beschreibung	Zeigt die Komponente obj auf dem Bildschirm an.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

sleep

Synopsis	integer function j_sleep (integer msec)
Argumente	msec integer
Beschreibung	Unterbricht die Ausführung der Applikation für msec Millisekunden.

start

Synopsis integer function **j_start** ()

Beschreibung Diese Funktion stellt eine Verbindung zu einem JAPI Kernel her, sofern auf der lokalen Maschine bereits ein Kernel läuft. Kann kein Kernel gefunden werden, so wird ein neuer Kernel gestartet und eine Verbindung aufgebaut. Schlägt dies auch fehl, so wird als Fehlercode `.false.` zurückgegeben

Beispiel

```
:
if( .not. j_start() ) then
  write(*,*) "can't connect to JAPI Kernel"
  goto 20
endif
:
```


sync

Synopsis procedure **j_sync** ()

Beschreibung Synchronisiert die Applikation mit dem JAPI Kernel.

textarea

Synopsis integer function **j_textarea** (integer obj , integer rows , integer columns)

Argumente obj integer
 rows integer
 columns integer

Beschreibung Diese Funktion erzeugt eine neue Textarea und gibt dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Die Textarea hat eine sichtbare Größe von **rows** Reihen und **columns** Spalten, und ist somit abhängig vom voreingestellten Font. Soll eine Größe in Pixeleinheiten eingestellt werden, so kann nachträglich die Funktion *j_setsize()* verwendet werden. Die Textarea besitzt voreingestellt zwei Scrollbars, die sich automatisch anpassen. Eine Textarea liefert einen Event, sobald sich der Textinhalt ändert.

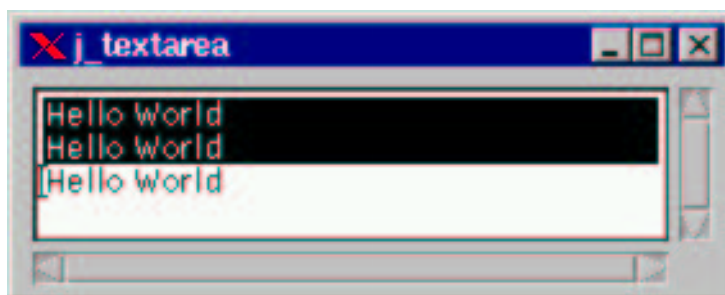
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame("j_textarea")
text  = j_textarea(frame,30,4)
:

```



textfield

Synopsis	integer function j_textfield (integer obj , integer columns)
Argumente	obj integer columns integer
Beschreibung	Diese Funktion erzeugt eine neues Textfield und gibt dessen Eventnummer zurück. Im Fehlerfall wird <code>-1</code> zurückgegeben. Ein Textfield dient für alle einzeiligen Eingaben. Das Textfield hat eine sichtbare Größe von genau einer Reihe und columns Spalten, und ist somit abhängig vom voreingestellten Font. Soll eine Größe in Pixeleinheiten eingestellt werden, so kann nachträglich die Funktion j_setsize() verwendet werden. Ein Textfield liefert einen Event, sobald die Eingabe mit Return abgeschlossen wird.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame("j_textfield")
text  = j_textfield(frame,30)
:

```



translate

Synopsis procedure **j_translate** (integer obj , integer x , integer y)

Argumente obj integer
 x integer
 y integer

Beschreibung Setzt den Ursprung des Koordinatensystems auf (**x**, **y**).

Zielkomponenten Canvas, Image, Printer

vscrollbar

Synopsis integer function **j_vscrollbar** (integer obj)

Argumente obj integer

Beschreibung Diese Funktion erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.Im Fehlerfall wird -1 zurückgegeben. Ist die Zielkomponente **obj** vom Typ Scrollpane, so wird die Eventnummer der vorhandenen vertikalen Scrollbar zurückgegeben. Eine Scrollbar erzeugt einen Event, sobald der Schieber bewegt wird.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame, Scrollpane

Beispiel

```

:
scroll=j_vscrollbar(frame)
call j_setpos(scroll,120,40)
call j_setsize(scroll,20,100)
:

```



windowlistener

Synopsis	integer function j_windowlistener (integer window , integer kind)
Argumente	window integer kind integer
Beschreibung	<p>Bindet an die Komponente window einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für kind sind:</p> <ul style="list-style-type: none">• J_ACTIVATED : Der Componentlistener liefert einen Event, wenn die Komponente obj aktiviert wird.• J_DEACTIVATED : Der Componentlistener liefert einen Event, wenn die Komponente obj deaktiviert wird.• J_OPENED : Der Componentlistener liefert einen Event, wenn die Komponente obj geöffnet ist.• J_CLOSED : Der Componentlistener liefert einen Event, wenn die Komponente obj geschlossen ist.• J_ICONFIED : Der Componentlistener liefert einen Event, wenn die Komponente obj iconifiziert wird.• J_DEICONFIED : Der Componentlistener liefert einen Event, wenn die Komponente obj deiconifiziert wird.• J_CLOSING : Der Componentlistener liefert einen Event, wenn das close icon der Komponenten obj angeklickt wird.
Zielkomponenten	Window, Dialog, Frame

window

Synopsis integer function **j_window** (integer obj)

Argumente obj integer

Beschreibung Dies Funktion erzeugt ein neues Window und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgeliefert. Ein Window ist ein einfaches eigenständiges Fenster, das keinen Fensterahmen besitzt. Wie ein Dialog, kann ein Window keine Menüleiste besitzen. Als Zielobjekt dieser Funktion ist nur ein Frame zulässig. Ein Window wird erst angezeigt, wenn ein entsprechendes *j_show()* Kommando ausgeführt wurde. Ein Window liefert keinen Event.

Zielkomponenten Frame

Beispiel

```

:
window = j_window(frame)
label  = j_label(window,"Mouse pressed at ... ")
call j_setnamedcolorbg(label,J_YELLOW)
:

```

Mouse pressed at 108:179