



Reference Manual

Merten Joost

Inhaltsverzeichnis

I	Reference	5
1	Components	7
	Button	7
	Borderpanel	10
	Canvas	15
	Checkbox	20
	CheckmenuItem	23
	Choice	24
	Dialog	27
	Focuslistener	33
	Frame	34
	Helpmenu	40
	Hscrollbar	42
	Graphicbutton	45
	Graphiclabel	48
	Image	51
	KeyListener	54
	Label	55
	Led	58
	List	61
	Menu	65
	MenuItem	67
	Meter	68
	Mouselistener	71
	Panel	72
	Popupmenu	77
	Printer	79
	Progressbar	82
	Radiobutton	85
	Sevensegment	88
	Scrollpane	91
	Textarea	94
	Textfield	98
	Vscrollbar	102
	Window	105

2 Functions	111
additem	111
add	113
alertbox	114
appendtext	115
beep	116
borderpanel	117
button	118
canvas	119
checkbox	120
checkmenuitem	121
choicebox2	122
choicebox3	123
choice	124
cliprect	125
componentlistener	126
connect	127
delete	128
deselect	129
dialog	130
disable	131
dispose	132
drawarc	133
drawcircle	134
drawimagesource	135
drawimage	136
drawline	137
drawoval	138
drawpixel	139
drawpolygon	140
drawpolyline	141
drawrect	142
drawroundrect	143
drawscaledimage	144
drawstring	145
enable	146
filedialog	147
fileselector	148
fillarc	149
fillcircle	150
filloval	151
fillpolygon	152
fillrect	153
fillroundrect	154
focuslistener	155
frame	156
getaction	157

getcolumns	158
getcurpos	159
getfontascent	160
getfontheight	161
getheight	162
getimagesource	163
getimage	164
getinsets	165
getitemcount	167
getitem	168
getkeychar	169
getkeycode	170
getlayoutid	171
getlength	172
getmousebutton	173
getmousex	174
getmousey	175
getparentid	176
getparent	177
getrows	178
getscaledimage	179
getscreenheight	180
getscreenwidth	181
getselect	182
getselend	183
getselstart	184
getseltext	185
getstate	186
gettext	187
getvalue	188
getviewportheight	189
getviewportwidth	190
getwidth	191
getxpos	192
getypos	193
graphicbutton	194
graphiclabel	195
hasfocus	196
helpmenu	197
hide	198
hscrollbar	199
image	200
insert	201
inserttext	202
isparent	203
isresizable	204
isselect	205

isvisible	206
keylistener	207
label	208
led	209
line	210
list	212
loadimage	213
menubar	214
menuitem	215
menu	216
messagebox	217
meter	218
mouselistener	219
multiplemode	220
nextaction	221
pack	222
panel	223
popupmenu	224
printer	225
print	226
progressbar	227
quit	228
radiobutton	229
radiogroup	230
random	231
releaseall	232
release	233
removeall	234
removeitem	235
remove	236
replacetext	237
saveimage	238
scrollpane	239
selectall	240
select	241
selecttext	242
seperator	243
setalign	244
setblockinc	245
setborderlayout	246
setborderpos	247
setcolorbg	248
setcolor	249
setcolumns	250
setcurpos	251
setcursor	252
setdebug	253

setechochar	254
seteditable	255
setfixlayout	256
setflowfill	257
setflowlayout	258
setfocus	259
setFontname	260
setfontsize	261
setFontstyle	262
setFont	263
setgridlayout	265
sethgap	266
seticon	267
setimage	268
setinsets	269
setmax	270
setmin	271
setnamedcolorbg	272
setnamedcolor	273
setnolayout	274
setpos	275
setradiogroup	276
setresizable	277
setrows	278
setshortcut	279
setsize	280
setslidesize	281
setstate	282
settext	283
setunitinc	284
setvalue	285
setvgap	286
setxor	287
sevensegment	288
showpopup	289
show	290
sleep	291
start	292
sync	293
textarea	294
textfield	295
translate	296
vscrollbar	297
windowlistener	298
window	299

Teil I

Reference

Kapitel 1

Components

Button

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

	Liefert die totale Höhe des aktuellen Fonts des Button obj in Bildschirmpixeln zurück.
j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe des Button obj zurück.
j_getlength	<i>int j_getlength (int obj);</i> Liefert die Länge des Textinhaltes den Button zurück.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> Liefert den Textinhalt des Button .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Button obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Button obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Button obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt den Button obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an den Button obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Button .

j_release	<i>void j_release (int obj);</i> Gibt den Button obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert den Button obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Button obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Button obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Button obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Button obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Button obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Button obj auf style .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt den Button obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Button obj auf width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Ändert den Textinhalt des Button obj auf str
j_show	<i>void j_show (int obj);</i> Zeigt den Button obj auf dem Bildschirm an.

Borderpanel

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

j_getfontheight	<i>int j_getfontheight (int obj);</i> Liefert die totale Höhe des aktuellen Fonts des Borderpanel obj in Bildschirm-pixeln zurück.
j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe des Borderpanel obj zurück.
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Liefert die angeforderte Breite des inneren Randes dem Borderpanel zurück.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird <code>-1</code> zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird <code>-1</code> zurückgegeben.
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Borderpanel obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Borderpanel obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Borderpanel obj zurück.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>void j_hide (int obj);</i> Zeigt das Borderpanel obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert <code>J_TRUE</code> zurück, falls das cont die umgebende Komponente von obj ist, sonst <code>J_FALSE</code>
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert <code>J_TRUE</code> zurück, falls obj sichtbar ist, sonst <code>J_FALSE</code>
j_keylistener	<i>int j_keylistener (int obj);</i>

Bindet an das Borderpanel **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.

j_label	<i>int j_label (int obj , char* label);</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>int j_led (int obj , int style , int color);</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>int j_list (int obj , int rows);</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>int j_meter (int obj , char* title);</i> Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung title .
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Bindet an das Borderpanel obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_pack	<i>void j_pack (int obj);</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>int j_panel (int obj);</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Borderpanel .
j_progressbar	<i>int j_progressbar (int obj , int orient);</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>void j_releaseall (int obj);</i> Gibt alle Elemente aus dem Borderpanel obj frei.
j_release	<i>void j_release (int obj);</i> Gibt das Borderpanel obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>int j_scrollpane (int obj);</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>void j_setalign (int obj , int align);</i>

Setzt die Ausrichtung innerhalb des Borderpanel **obj** auf **align**. Ein Flowlayout Manager wird vorausgesetzt.

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

j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Entfernt den aktuellen Layout Manager aus dem Borderpanel obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt das Borderpanel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Borderpanel obj auf width and height .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Setzt den vertikalen Abstand innerhalb des Borderpanel obj auf vgap Pixel.
j_sevensegment	<i>int j_sevensegment (int obj , int color);</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>void j_show (int obj);</i> Zeigt das Borderpanel obj auf dem Bildschirm an.
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

Canvas

j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Erzeugt einen neuen Canvas mit der Breite width Pixel und einer Höhe von height Pixel.
j_add	<i>void j_add (int obj , int cont);</i> Fügt den Canvas obj in den Container cont ein.
j_cliprect	<i>void j_cliprect (int obj , int x , int y , int width , int height);</i> Setzt den Clipping Bereich auf das spezifizierte Rechteck (x , y , width , height).
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Bindet an den Canvas obj einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_disable	<i>void j_disable (int obj);</i> disabled den Canvas obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>void j_dispose (int obj);</i> der Canvas obj wird mit allen Recourcen gelöscht.
j_drawarc	<i>void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> 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	<i>void j_drawcircle (int obj , int x , int y , int r);</i> Zeichnet einen Kreis mit dem Mittelpunkt (x,y) und dem Radius r .
j_drawimage	<i>void j_drawimage (int obj , int image , int x , int y);</i> Kopiert das Image mit der Eventnummer image an die Position (x , y).
j_drawimagesource	<i>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> 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	<i>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</i> Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (x1,y1) nach (x2,y2)
j_drawoval	<i>void j_drawoval (int obj , int x , int y , int rx , int ry);</i>

	Zeichnet ein Oval mit dem Mittelpunkt (x,y) und den Hauptradien rx und ry .
j_drawpixel	<i>void j_drawpixel (int obj , int x , int y);</i> Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (x,y) .
j_drawpolygon	<i>void j_drawpolygon (int obj , int len , int* x , int* y);</i> Zeichnet ein Polygon durch die Punkte, die durch die Arrays x und y gegeben sind.
j_drawpolyline	<i>void j_drawpolyline (int obj , int len , int* x , int* y);</i> Zeichnet eine Polylinie durch die Punkte, die durch die Arrays x und y gegeben sind.
j_drawrect	<i>void j_drawrect (int obj , int x , int y , int width , int height);</i> Zeichnet ein Rechteck an die Position (x,y) mit der Breite width und der Höhe height .
j_drawroundrect	<i>void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int 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_drawscaledimage	<i>void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int 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>void j_drawstring (int obj , int x , int y , char* str);</i> Zeichnet den String str an die durch die Parameter (x,y) festgelegte Koordinate.
j_enable	<i>void j_enable (int obj);</i> enabled den Canvas obj .
j_fillarc	<i>void j_fillarc (int obj , int x , int y , int rx , int ry , int arc1 , int 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>void j_fillcircle (int obj , int x , int y , int r);</i> Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (x,y) und dem Radius r .
j_filloval	<i>void j_filloval (int obj , int x , int y , int rx , int ry);</i> Zeichnet einen gefüllten Oval mit dem Mittelpunkt (x,y) und den Hauptradien rx und ry .
j_fillpolygon	<i>void j_fillpolygon (int obj , int len , int* x , int* y);</i> Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays x und y gegeben sind.

j_fillrect	<i>void j_fillrect (int obj , int x , int y , int width , int height);</i> Zeichnet ein gefülltes Rechteck an die Position (x,y) mit der Breite width und der Höhe height .
j_fillroundrect	<i>void j_fillroundrect (int obj , int x , int y , int width , int height , int arcx , int 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_focuslistener	<i>int j_focuslistener (int obj);</i> Bindet an den Canvas obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Liefert die Oberlänge (Ascent) des aktuellen Fonts des Canvas obj in Bildschirmpixeln zurück.
j_getfontheight	<i>int j_getfontheight (int obj);</i> Liefert die totale Höhe des aktuellen Fonts des Canvas obj in Bildschirmpixeln zurück.
j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe des Canvas obj zurück.
j_getimage	<i>int j_getimage (int obj);</i> Kopiert den Inhalt dem Canvas obj in ein Image und liefert dessen Eventnummer zurück.
j_getimagesource	<i>int j_getimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> Die Funktion liefert einen Ausschnitt vom Punkt (x, y) der Breite w und der Höhe h zurck. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays r,g,b abgelegt.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getscaledimage	<i>int j_getscaledimage (int obj , int x , int y , int sw , int sh , int tw , int 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>int j_getwidth (int obj);</i> Liefert die Breite des Canvas obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Canvas obj zurück.

j_gettypos	<i>int j_gettypos (int obj);</i> Liefert die aktuelle vertikale Position des Canvas obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt den Canvas obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an den Canvas obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popuptmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Canvas .
j_release	<i>void j_release (int obj);</i> Gibt den Canvas obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert den Canvas obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Canvas obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Canvas obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Canvas obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Canvas obj entsprechend der Parameterliste.

j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Canvas obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Canvas obj auf style .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt den Canvas obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Canvas obj auf width and height .
j_setxor	<i>void j_setxor (int obj , int bool);</i> Schaltet den Painting Mode auf XOR, falls bool = J_TRUE ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
j_show	<i>void j_show (int obj);</i> Zeigt den Canvas obj auf dem Bildschirm an.
j_translate	<i>void j_translate (int obj , int x , int y);</i> Setzt den Ursprung des Koordinatensystems auf (x, y).

Checkbox

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

j_gettext	<i>char* j_gettext (int obj , char* str);</i> Liefert den Textinhalt der Checkbox .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite der Checkbox obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der Checkbox obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der Checkbox obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die Checkbox obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die Checkbox obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Checkbox .
j_release	<i>void j_release (int obj);</i> Gibt die Checkbox obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die Checkbox obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i>

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

Checkmenuitem

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

Choice

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

j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselect	<i>int j_getselect (int obj);</i> Liefert den Index des selectierten Items der Choice .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite der Choice obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der Choice obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der Choice obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die Choice obj nicht mehr auf dem Bildschirm an.
j_insert	<i>int j_insert (int obj , int pos , char* label);</i> fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die Choice obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Choice .
j_release	<i>void j_release (int obj);</i> Gibt die Choice obj aus dem umgebenden Container Element frei.
j_removeall	<i>int j_removeall (int obj);</i> entfernt alle Items aus der Choice .
j_removeitem	<i>int j_removeitem (int obj , char* item);</i> entfernt den ersten Eintrag item aus der Choice .

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

Dialog

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

Liefert die totale Höhe des aktuellen Fonts des Dialog **obj** in Bildschirmpixeln zurück.

j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe des Dialog obj zurück.
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Liefert die angeforderte Breite des inneren Randes dem Dialog zurück.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getlength	<i>int j_getlength (int obj);</i> Liefert die Länge des Textinhaltes den Dialog zurück.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> Liefert den Textinhalt des Dialog .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Dialog obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Dialog obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Dialog obj zurück.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>void j_hide (int obj);</i> Zeigt den Dialog obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE

j_isresizable	<i>int j_isresizable (int obj);</i> Liefert J_FALSE zurueck, falls der Dialog eine fixe Größe besitzt, J_TRUE sonst.
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an den Dialog obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>int j_label (int obj , char* label);</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>int j_led (int obj , int style , int color);</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>int j_list (int obj , int rows);</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>int j_meter (int obj , char* title);</i> Erzeugt ein analoges Anzeigeeinstrument mit der Beschriftung title .
j_mouselistener	<i>int j_mouselistener (int obj , int 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>void j_pack (int obj);</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>int j_panel (int obj);</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Dialog .
j_progressbar	<i>int j_progressbar (int obj , int orient);</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>void j_releaseall (int obj);</i> Gibt alle Elemente aus dem Dialog obj frei.
j_release	<i>void j_release (int obj);</i>

Gibt den Dialog **obj** aus dem umgebenden Container Element frei.

j_scrollpane	<i>int j_scrollpane (int obj);</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>void j_setalign (int obj , int align);</i> Setzt die Ausrichtung innerhalb des Dialog obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Integriert einen BorderLayout Manager in den Dialog obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert den Dialog obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Dialog obj auf cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Integriert einen Fixlayout Manager in den Dialog obj (voreingestellte Layout-manager).
j_setflowfill	<i>void j_setflowfill (int obj , int 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>void j_setflowlayout (int obj , int align);</i> Integriert einen Flowlayout Manager in den Dialog obj mit der Ausrichtung align .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Dialog obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Dialog obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Dialog obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Dialog obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Dialog obj auf style .

j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Integriert einen Gridlayout Manager in den Dialog obj mit row Reihen und col Spalten.
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Setzt den horizontalen Abstand innerhalb des Dialog obj auf hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Entfernt den aktuellen Layout Manager aus dem Dialog obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt den Dialog obj zur angegebenen Koordinate (xpos,ypos) an.
j_setresizable	<i>void j_setresizable (int obj , int resizable);</i> Ist resizable J_FALSE , kann die Größe des Dialog nicht vom Benutzer verändert werden.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Dialog obj auf width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Ändert den Textinhalt des Dialog obj auf str
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Setzt den vertikalen Abstand innerhalb des Dialog obj auf vgap Pixel.
j_sevenssegment	<i>int j_sevenssegment (int obj , int color);</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>void j_show (int obj);</i> Zeigt den Dialog obj auf dem Bildschirm an.
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>int j_windowlistener (int window , int kind);</i>

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	<i>int j_focuslistener (int obj);</i> Bindet an den Focuslistener obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
j_dispose	<i>void j_dispose (int obj);</i> der Focuslistener obj wird mit allen Recourcen gelöscht.
j_hasfocus	<i>int j_hasfocus (int obj);</i> Liefert J_TRUE zurück, falls der Focuslistener den Focus besitzt, sonst J_FALSE .

Frame

j.frame	<i>int j_frame (char* label);</i> Erzeugt einen Frame mit der Titelzeile label .
j.add	<i>void j_add (int obj , int cont);</i> Fügt den Frame obj in den Container cont ein.
j.alertbox	<i>void j_alertbox (int obj , char* title , char* text , char* button);</i> Öffnet eine Alertbox mit der Überschrift title , Textinhalt text und einem Button.
j.borderpanel	<i>int j_borderpanel (int obj , int type);</i> Erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück.
j.button	<i>int j_button (int obj , char* label);</i> Erzeugt einen neuen Button mit dem Textinhalt label
j.canvas	<i>int j_canvas (int obj , int width , int height);</i> Erzeugt einen neuen Canvas mit der Breite width Pixel und einer Höhe von height Pixel.
j.checkbox	<i>int j_checkbox (int obj , char* label);</i> Erzeugt eine Checkbox mit dem Textinhalt label und liefert dessen Eventnummer zurück.
j.choicebox2	<i>void j_choicebox2 (int obj , char* title , char* text , char* button1 , char* button2);</i> Öffnet eine Choicebox mit der Überschrift title , Textinhalt text und zwei Buttons.
j.choicebox3	<i>void j_choicebox3 (int obj , char* title , char* text , char* button1 , char* button2 , char* button3);</i> Öffnet eine Choicebox mit der Überschrift title , Textinhalt text und drei Buttons.
j.choice	<i>int j_choice (int obj);</i> Erzeugt eine neue Choice und gibt deren Eventnummer zurück.
j.componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Bindet an den Frame obj einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j.dialog	<i>int j_dialog (int obj , char* label);</i> Erzeugt ein neues Dialogwindow mit dem Titel label und liefert dessen Eventnummer zurück.
j.disable	<i>void j_disable (int obj);</i>

disabled den Frame **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.

j_dispose	<i>void j_dispose (int obj);</i> der Frame obj wird mit allen Recourcen gelöscht.
j_enable	<i>void j_enable (int obj);</i> enabled den Frame obj .
j_filedialog	<i>char* j_filedialog (int frame , char* title , char* directory , char* filename);</i> Öffnet eine Filedialogbox in dem angegebenen directory mit dem Title title und liefert den ausgewählten Dateiname in filename zurück.
j_fileselector	<i>char* j_fileselector (int frame , char* title , char* filter , char* filename);</i> Öffnet eine Fileslectorbox mit dem voreingestellten filename und dem Title title und liefert den ausgewählten Dateiname in filename zurück.
j_focuslistener	<i>int j_focuslistener (int obj);</i> Bindet an den Frame obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Liefert die Oberlänge (Ascent) des aktuellen Fonts des Frame obj in Bildschirmpixeln zurück.
j_getfontheight	<i>int j_getfontheight (int obj);</i> Liefert die totale Höhe des aktuellen Fonts des Frame obj in Bildschirmpixeln zurück.
j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe des Frame obj zurück.
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Liefert die angeforderte Breite des inneren Randes dem Frame zurück.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getlength	<i>int j_getlength (int obj);</i> Liefert die Länge des Textinhaltes den Frame zurück.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> Liefert den Textinhalt des Frame .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Frame obj zurück.

j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Frame obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Frame obj zurück.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>void j_hide (int obj);</i> Zeigt den Frame obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isresizable	<i>int j_isresizable (int obj);</i> Liefert J_FALSE zurueck, falls der Frame eine fixe Größe besitzt, J_TRUE sonst.
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an den Frame obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_label	<i>int j_label (int obj , char* label);</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>int j_led (int obj , int style , int color);</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>int j_list (int obj , int rows);</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_menubar	<i>int j_menubar (int obj);</i> Erzeugt eine neue Menubar in obj und liefert dessen Eventnummer zurück.
j_messagebox	<i>void j_messagebox (int obj , char* title , char* text);</i> Öffnet eine MessageBox mit der Überschrift title und dem Textinhalt text

j_meter	<i>int j_meter (int obj , char* title);</i> Erzeugt ein analoges Anzeigeinstrument mit der Beschriftung title .
j_mouselistener	<i>int j_mouselistener (int obj , int 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>void j_pack (int obj);</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>int j_panel (int obj);</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popuptmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_printer	<i>int j_printer (int frame);</i> Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Frame .
j_progressbar	<i>int j_progressbar (int obj , int orient);</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>void j_releaseall (int obj);</i> Gibt alle Elemente aus dem Frame obj frei.
j_release	<i>void j_release (int obj);</i> Gibt den Frame obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>int j_scrollpane (int obj);</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>void j_setalign (int obj , int align);</i> Setzt die Ausrichtung innerhalb des Frame obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Integriert einen BorderLayout Manager in den Frame obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert den Frame obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert

j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Frame obj auf cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Integriert einen Fixlayout Manager in den Frame obj (voreingestellte Layout-manager).
j_setflowfill	<i>void j_setflowfill (int obj , int 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>void j_setflowlayout (int obj , int align);</i> Integriert einen Flowlayout Manager in den Frame obj mit der Ausrichtung align .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Frame obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Frame obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Frame obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Frame obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Frame obj auf style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Integriert einen Gridlayout Manager in den Frame obj mit row Reihen und col Spalten.
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Setzt den horizontalen Abstand innerhalb des Frame obj auf hgap Pixel.
j_seticon	<i>void j_seticon (int frame , int icon);</i> Ordnet dem frame das Icon icon zu. Eine iconisierte Darstellung wird nicht von allen Plattformen unterstützt.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Setzt den inneren Rahmen auf die angegebenen Größen.
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .

j_setnolayout	<i>void j_setnolayout (int obj);</i> Entfernt den aktuellen Layout Manager aus dem Frame obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt den Frame obj zur angegebenen Koordinate (xpos,ypos) an.
j_setresizable	<i>void j_setresizable (int obj , int resizable);</i> Ist resizable J.FALSE , kann die Größe des Frame nicht vom Benutzer verändert werden.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Frame obj auf width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Ändert den Textinhalt des Frame obj auf str
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Setzt den vertikalen Abstand innerhalb des Frame obj auf vgap Pixel.
j_sevensegment	<i>int j_sevensegment (int obj , int color);</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>void j_show (int obj);</i> Zeigt den Frame obj auf dem Bildschirm an.
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>int j_windowlistener (int window , int 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	<i>int j_window (int obj);</i> Erzeugt ein neues Window und liefert dessen Eventnummer zurück.

Helpmenu

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

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

Hscrollbar

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

j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite der Hscrollbar obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der Hscrollbar obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der Hscrollbar obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die Hscrollbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die Hscrollbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Hscrollbar .
j_release	<i>void j_release (int obj);</i> Gibt die Hscrollbar obj aus dem umgebenden Container Element frei.
j_setblockinc	<i>int j_setblockinc (int obj , int val);</i> Stellt die Blockänderung von der Hscrollbar auf den Wert von val ein.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die Hscrollbar obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i>

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

Graphicbutton

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

Liefert die Breite des Graphicbutton **obj** zurück.

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

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

Graphiclabel

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

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

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

Image

j_image	<i>int j_image (int width , int height);</i> Erzeugt ein neues (Memory) Image mit der Breite width Pixel und einer Höhe von height Pixel.
j_cliprect	<i>void j_cliprect (int obj , int x , int y , int width , int height);</i> Setzt den Clipping Bereich auf das spezifizierte Rechteck (x , y , width , height).
j_dispose	<i>void j_dispose (int obj);</i> das Image obj wird mit allen Recourcen gelöscht.
j_drawarc	<i>void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> 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	<i>void j_drawcircle (int obj , int x , int y , int r);</i> Zeichnet einen Kreis mit dem Mittelpunkt (x , y) und dem Radius r .
j_drawimage	<i>void j_drawimage (int obj , int image , int x , int y);</i> Kopiert das Image mit der Eventnummer image an die Position (x , y).
j_drawimagesource	<i>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> 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	<i>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</i> Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (x1 , y1) nach (x2 , y2)
j_drawoval	<i>void j_drawoval (int obj , int x , int y , int rx , int ry);</i> Zeichnet ein Oval mit dem Mittelpunkt (x , y) und den Hauptradien rx und ry .
j_drawpixel	<i>void j_drawpixel (int obj , int x , int y);</i> Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (x , y).
j_drawpolygon	<i>void j_drawpolygon (int obj , int len , int* x , int* y);</i> Zeichnet ein Polygon durch die Punkte, die durch die Arrays x und y gegeben sind.
j_drawpolyline	<i>void j_drawpolyline (int obj , int len , int* x , int* y);</i>

Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.

- j_drawrect** *void j_drawrect (int obj , int x , int y , int width , int height);*
Zeichnet ein Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int 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** *void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);*
Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter (**sx**, **sy**, **sw**, **sh**) bestimmt wird an die Position (**tx**, **ty**) und skaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**.
- j_drawstring** *void j_drawstring (int obj , int x , int y , char* str);*
Zeichnet den String **str** an die durch die Parameter (**x,y**) festgelegte Koordinate.
- j_fillarc** *void j_fillarc (int obj , int x , int y , int rx , int ry , int arc1 , int 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** *void j_fillcircle (int obj , int x , int y , int r);*
Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (**x,y**) und dem Radius **r**.
- j_filloval** *void j_filloval (int obj , int x , int y , int rx , int ry);*
Zeichnet einen gefüllten Oval mit dem Mittelpunkt (**x,y**) und den Hauptradien **rx** und **ry**.
- j_fillpolygon** *void j_fillpolygon (int obj , int len , int* x , int* y);*
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_fillrect** *void j_fillrect (int obj , int x , int y , int width , int height);*
Zeichnet ein gefülltes Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_fillroundrect** *void j_fillroundrect (int obj , int x , int y , int width , int height , int arcx , int 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_getheight** *int j_getheight (int obj);*
Liefert die Höhe des Image **obj** zurück.
- j_getimage** *int j_getimage (int obj);*

Kopiert den Inhalt dem Image **obj** in ein Image und liefert dessen Eventnummer zurück.

j_getimagesource *int j_getimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* 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_getscaledimage *int j_getscaledimage (int obj , int x , int y , int sw , int sh , int tw , int th);*

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 *int j_getwidth (int obj);*

Liefert die Breite des Image **obj** zurück.

j_print *void j_print (int obj);*

Druckt den Inhalt des Image .

j_setxor *void j_setxor (int obj , int bool);*

Schaltet den Painting Mode auf XOR, falls bool = J_TRUE ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.

j_translate *void j_translate (int obj , int x , int y);*

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

KeyListener

j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an den Keylistener obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_dispose	<i>void j_dispose (int obj);</i> der Keylistener obj wird mit allen Recourcen gelöscht.
j_getkeychar	<i>int j_getkeychar (int obj);</i> Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.
j_getkeycode	<i>int j_getkeycode (int obj);</i> Liefert den Keycode, der zuletzt gedrückten Taste zurück.

Label

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

Liefert die Breite des Label **obj** zurück.

j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Label obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Label obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt das Label obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an das Label obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Label .
j_release	<i>void j_release (int obj);</i> Gibt das Label obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert das Label obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Label obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Label obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i>

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

Led

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

Liefert die Breite der LED **obj** zurück.

j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der LED obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der LED obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die LED obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die LED obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der LED .
j_release	<i>void j_release (int obj);</i> Gibt die LED obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die LED obj an die angegebene Position pos . Im umgebenden Container muß einen Borderlayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor der LED obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus der LED obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i>

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

List

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

j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselect	<i>int j_getselect (int obj);</i> Liefert den Index des selectierten Items der List .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite der List obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der List obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der List obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die List obj nicht mehr auf dem Bildschirm an.
j_insert	<i>int j_insert (int obj , int pos , char* label);</i> fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_iselect	<i>int j_iselect (int obj , int item);</i> Liefert J_TRUE zurück, falls das Item mit dem Index item selectiert ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die List obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_multiplemode (int obj , int bool);</i> die List erlaubt mehrfache Selectierung, falls bool J_TRUE ist.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popuptmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i>

Druckt den Inhalt der List .

j_release	<i>void j_release (int obj);</i> Gibt die List obj aus dem umgebenden Container Element frei.
j_removeall	<i>int j_removeall (int obj);</i> entfernt alle Items aus der List .
j_removeitem	<i>int j_removeitem (int obj , char* item);</i> entfernt den ersten Eintrag item aus der List .
j_remove	<i>int j_remove (int obj , int item);</i> entfernt das Item mit dem Index item aus der List .
j_select	<i>int j_select (int obj , int item);</i> selectiert das Item mit dem Index item .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die List obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor der List obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus der List obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp der List obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font der List obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße der List obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle der List obj auf style .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i>

Verschiebt die List **obj** zur angegebenen Koordinate (**xpos,ypos**) an.

j_setsize

void j_setsize (int obj , int width , int height);

Setzt Breite und Höhe der List **obj** auf **width** and **height**.

j_show

void j_show (int obj);

Zeigt die List **obj** auf dem Bildschirm an.

Menu

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

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

Menuitem

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

Meter

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

j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Anzeigeinstrument obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Anzeigeinstrument obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt das Anzeigeinstrument obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an das Anzeigeinstrument obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Anzeigeinstrument .
j_release	<i>void j_release (int obj);</i> Gibt das Anzeigeinstrument obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert das Anzeigeinstrument obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Anzeigeinstrument obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Anzeigeinstrument obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i>

	Setzt den Fonttyp des Anzeigeinstrument obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Anzeigeinstrument obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Anzeigeinstrument obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Anzeigeinstrument obj auf style .
j_setmax	<i>int j_setmax (int obj , int val);</i> Stellt die obere Grenze dem Anzeigeinstrument auf den Wert von val ein.
j_setmin	<i>int j_setmin (int obj , int val);</i> Stellt die untere Grenze dem Anzeigeinstrument auf den Wert von val ein.
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt das Anzeigeinstrument obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Anzeigeinstrument obj auf width and height .
j_setvalue	<i>void j_setvalue (int obj , int val);</i> Stellt den aktuellen Wert dem Anzeigeinstrument obj auf val .
j_show	<i>void j_show (int obj);</i> Zeigt das Anzeigeinstrument obj auf dem Bildschirm an.

Mouselistener

j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Bindet an den Mouselistener obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_dispose	<i>void j_dispose (int obj);</i> der Mouselistener obj wird mit allen Recourcen gelöscht.
j_getmousebutton	<i>int j_getmousebutton (int mouselistener);</i> Liefert die letzte benutzte Mousetaste zurück.
j_getmousex	<i>int j_getmousex (int mouselistener);</i> Liefert die aktuelle horizontale Position der Mouse zurück.
j_getmousey	<i>int j_getmousey (int mouselistener);</i> Liefert die aktuelle vertikale Position der Mouse zurück.

Panel

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

	Liefert die totale Höhe des aktuellen Fonts des Panel obj in Bildschirmpixeln zurück.
j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe des Panel obj zurück.
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Liefert die angeforderte Breite des inneren Randes dem Panel zurück.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Panel obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Panel obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Panel obj zurück.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch filename spezifiziert wird.
j_hide	<i>void j_hide (int obj);</i> Zeigt das Panel obj nicht mehr auf dem Bildschirm an.
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an das Panel obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.

j_label	<i>int j_label (int obj , char* label);</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>int j_led (int obj , int style , int color);</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>int j_list (int obj , int rows);</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>int j_meter (int obj , char* title);</i> Erzeugt ein analoges Anzeigeelement mit der Beschriftung title .
j_mouselistener	<i>int j_mouselistener (int obj , int 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>void j_pack (int obj);</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>int j_panel (int obj);</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Panel .
j_progressbar	<i>int j_progressbar (int obj , int orient);</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>void j_releaseall (int obj);</i> Gibt alle Elemente aus dem Panel obj frei.
j_release	<i>void j_release (int obj);</i> Gibt das Panel obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>int j_scrollpane (int obj);</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>void j_setalign (int obj , int align);</i> Setzt die Ausrichtung innerhalb des Panel obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i>

	Integriert einen BorderLayout Manager in das Panel obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert das Panel obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Panel obj auf cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Integriert einen Fixlayout Manager in das Panel obj (voreingestellte Layout-manager).
j_setflowfill	<i>void j_setflowfill (int obj , int 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>void j_setflowlayout (int obj , int align);</i> Integriert einen Flowlayout Manager in das Panel obj mit der Ausrichtung align .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Panel obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Panel obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Panel obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Panel obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Panel obj auf style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Integriert einen Gridlayout Manager in das Panel obj mit row Reihen und col Spalten.
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Setzt den horizontalen Abstand innerhalb des Panel obj auf hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Setzt den inneren Rahmen auf die angegebenen Größen.

j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Entfernt den aktuellen Layout Manager aus dem Panel obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt das Panel obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Panel obj auf width and height .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Setzt den vertikalen Abstand innerhalb des Panel obj auf vgap Pixel.
j_sevenssegment	<i>int j_sevenssegment (int obj , int color);</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>void j_show (int obj);</i> Zeigt das Panel obj auf dem Bildschirm an.
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.

PopupMenu

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

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

j_showpopup

void j_showpopup (int obj , int xpos , int ypos);

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

Printer

j_printer	<i>int j_printer (int frame);</i> Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
j_cliprect	<i>void j_cliprect (int obj , int x , int y , int width , int height);</i> Setzt den Clipping Bereich auf das spezifizierte Rechteck (x , y , width , height).
j_dispose	<i>void j_dispose (int obj);</i> der Printer obj wird mit allen Recourcen gelöscht.
j_drawarc	<i>void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> 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	<i>void j_drawcircle (int obj , int x , int y , int r);</i> Zeichnet einen Kreis mit dem Mittelpunkt (x,y) und dem Radius r .
j_drawimage	<i>void j_drawimage (int obj , int image , int x , int y);</i> Kopiert das Image mit der Eventnummer image an die Position (x , y).
j_drawimagesource	<i>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> 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	<i>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</i> Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von (x1,y1) nach (x2,y2)
j_drawoval	<i>void j_drawoval (int obj , int x , int y , int rx , int ry);</i> Zeichnet einen Oval mit dem Mittelpunkt (x,y) und den Hauptradien rx und ry .
j_drawpixel	<i>void j_drawpixel (int obj , int x , int y);</i> Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (x,y).
j_drawpolygon	<i>void j_drawpolygon (int obj , int len , int* x , int* y);</i> Zeichnet ein Polygon durch die Punkte, die durch die Arrays x und y gegeben sind.
j_drawpolyline	<i>void j_drawpolyline (int obj , int len , int* x , int* y);</i> Zeichnet eine Polylinie durch die Punkte, die durch die Arrays x und y gegeben sind.

j_drawrect	<pre>void j_drawrect (int obj , int x , int y , int width , int height);</pre> <p>Zeichnet ein Rechteck an die Position (x,y) mit der Breite width und der Höhe height.</p>
j_drawroundrect	<pre>void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);</pre> <p>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.</p>
j_drawscaledimage	<pre>void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);</pre> <p>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.</p>
j_drawstring	<pre>void j_drawstring (int obj , int x , int y , char* str);</pre> <p>Zeichnet den String str an die durch die Parameter (x,y) festgelegte Koordinate.</p>
j_fillarc	<pre>void j_fillarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</pre> <p>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.</p>
j_fillcircle	<pre>void j_fillcircle (int obj , int x , int y , int r);</pre> <p>Zeichnet einen gefüllten Kreis mit dem Mittelpunkt (x,y) und dem Radius r.</p>
j_filloval	<pre>void j_filloval (int obj , int x , int y , int rx , int ry);</pre> <p>Zeichnet einen gefüllten Oval mit dem Mittelpunkt (x,y) und den Hauptradien rx und ry.</p>
j_fillpolygon	<pre>void j_fillpolygon (int obj , int len , int* x , int* y);</pre> <p>Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays x und y gegeben sind.</p>
j_fillrect	<pre>void j_fillrect (int obj , int x , int y , int width , int height);</pre> <p>Zeichnet ein gefülltes Rechteck an die Position (x,y) mit der Breite width und der Höhe height.</p>
j_fillroundrect	<pre>void j_fillroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);</pre> <p>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.</p>
j_print	<pre>void j_print (int obj);</pre> <p>Druckt den Inhalt des Printer .</p>
j_setxor	<pre>void j_setxor (int obj , int bool);</pre> <p>Schaltet den Painting Mode auf XOR, falls bool = J_TRUE ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.</p>

j_translate*void j_translate (int obj , int x , int y);*Setzt den Ursprung des Koordinatensystems auf (**x**, **y**).

Progressbar

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

j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der Progressbar obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der Progressbar obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die Progressbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die Progressbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Bindet an die Progressbar obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Progressbar .
j_release	<i>void j_release (int obj);</i> Gibt die Progressbar obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die Progressbar obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor der Progressbar obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus der Progressbar obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i>

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

Radiobutton

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

Liefert J_TRUE , falls der Radiobutton selektiert ist, sonst J_FALSE .

j_gettext	<i>char* j_gettext (int obj , char* str);</i> Liefert den Textinhalt des Radiobutton .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Radiobutton obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Radiobutton obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Radiobutton obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt den Radiobutton obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an den Radiobutton obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Radiobutton .
j_release	<i>void j_release (int obj);</i> Gibt den Radiobutton obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert den Radiobutton obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert

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

Sevenssegment

j_sevenssegment	<i>int j_sevenssegment (int obj , int color);</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_add	<i>void j_add (int obj , int cont);</i> Fügt die Siebensegment-Anzeige obj in den Container cont ein.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> 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	<i>void j_disable (int obj);</i> disabled die Siebensegment-Anzeige obj , sodaß diese auf keine Benutzereingaben mehr reagiert.
j_dispose	<i>void j_dispose (int obj);</i> die Siebensegment-Anzeige obj wird mit allen Recourcen gelöscht.
j_enable	<i>void j_enable (int obj);</i> enabled die Siebensegment-Anzeige obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Bindet an die Siebensegment-Anzeige obj einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Liefert die Oberlänge (Ascent) des aktuellen Fonts der Siebensegment-Anzeige obj in Bildschirmpixeln zurück.
j_getfontheight	<i>int j_getfontheight (int obj);</i> Liefert die totale Höhe des aktuellen Fonts der Siebensegment-Anzeige obj in Bildschirmpixeln zurück.
j_getheight	<i>int j_getheight (int obj);</i> Liefert die Höhe der Siebensegment-Anzeige obj zurück.
j_getparentid	<i>int j_getparentid (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite der Siebensegment-Anzeige obj zurück.

j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der Siebensegment-Anzeige obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der Siebensegment-Anzeige obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die Siebensegment-Anzeige obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die Siebensegment-Anzeige obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> 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	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Siebensegment-Anzeige .
j_release	<i>void j_release (int obj);</i> Gibt die Siebensegment-Anzeige obj aus dem umgebenden Container Element frei.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die Siebensegment-Anzeige obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor der Siebensegment-Anzeige obj auf cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus der Siebensegment-Anzeige obj zu.

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

Scrollpane

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

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

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

Textarea

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

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

	Bindet an die Textarea obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Bindet an die Textarea obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Textarea .
j_release	<i>void j_release (int obj);</i> Gibt die Textarea obj aus dem umgebenden Container Element frei.
j_replacetext	<i>void j_replacetext (int obj , char* text , int start , int end);</i> Ersetzt den Text zwischen Position start bis end durch text .
j_selectall	<i>void j_selectall (int obj);</i> Selectiert den gesamten Textinhalt der Textarea .
j_selecttext	<i>void j_selecttext (int obj , int start , int end);</i> Selectiert den Text von Position start bis end .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die Textarea obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcolumns	<i>void j_setcolumns (int obj , int columns);</i> Setzt die Anzahl Spalten von obj auf columns .
j_setcurpos	<i>void j_setcurpos (int obj , int pos);</i> Setzt den Text Cursor auf die Position pos .
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor der Textarea obj auf cursor .
j_seteditable	<i>void j_seteditable (int obj , int bool);</i> Ist bool J.TRUE , so ist die Textarea editierbar, ansonsten ist die Textarea nur lesbar für den Benutzer.
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus der Textarea obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i>

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

Textfield

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

	Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>int j_getparent (int obj);</i> Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getselend	<i>int j_getselend (int obj);</i> Liefert die Endposition des selectierten Textes.
j_getselstart	<i>int j_getselstart (int obj);</i> Liefert die Anfangsposition des selectierten Textes.
j_getseltext	<i>char* j_getseltext (int obj , char* text);</i> Liefert den selectierten Text des Textfield obj zurück.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> Liefert den Textinhalt des Textfield .
j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite des Textfield obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position des Textfield obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position des Textfield obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt das Textfield obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an das Textfield obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i>

Druckt den Inhalt des Textfield .

j_release	<i>void j_release (int obj);</i> Gibt das Textfield obj aus dem umgebenden Container Element frei.
j_selectall	<i>void j_selectall (int obj);</i> Selectiert den gesamten Textinhalt des Textfield .
j_selecttext	<i>void j_selecttext (int obj , int start , int end);</i> Selectiert den Text von Position start bis end .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert das Textfield obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcolumns	<i>void j_setcolumns (int obj , int columns);</i> Setzt die Anzahl Spalten von obj auf columns .
j_setcurpos	<i>void j_setcurpos (int obj , int pos);</i> Setzt den Text Cursor auf die Position pos .
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Textfield obj auf cursor .
j_setechochar	<i>void j_setechochar (int obj , char chr);</i> Bestimmt das Zeichen chr als Echo Zeichen für alle Benutzereingaben in das Textfield .
j_seteditable	<i>void j_seteditable (int obj , int bool);</i> Ist bool J_TRUE , so ist das Textfield editierbar, ansonsten ist das Textfield nur lesbar für den Benutzer.
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Textfield obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Textfield obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Textfield obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Textfield obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Textfield obj auf style .

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

Vscrollbar

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

j_getwidth	<i>int j_getwidth (int obj);</i> Liefert die Breite der Vscrollbar obj zurück.
j_getxpos	<i>int j_getxpos (int obj);</i> Liefert die aktuelle horizontale Position der Vscrollbar obj zurück.
j_getypos	<i>int j_getypos (int obj);</i> Liefert die aktuelle vertikale Position der Vscrollbar obj zurück.
j_hide	<i>void j_hide (int obj);</i> Zeigt die Vscrollbar obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>int j_isvisible (int obj);</i> Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>int j_keylistener (int obj);</i> Bindet an die Vscrollbar obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>int j_mouselistener (int obj , int 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>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt der Vscrollbar .
j_release	<i>void j_release (int obj);</i> Gibt die Vscrollbar obj aus dem umgebenden Container Element frei.
j_setblockinc	<i>int j_setblockinc (int obj , int val);</i> Stellt die Blockänderung von der Vscrollbar auf den Wert von val ein.
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert die Vscrollbar obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i>

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

Window

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

Liefert die totale Höhe des aktuellen Fonts des Window **obj** in Bildschirmpixeln zurück.

- j_getheight** *int j_getheight (int obj);*
Liefert die Höhe des Window **obj** zurück.
- j_getinsets** *int j_getinsets (int obj , int side);*
Liefert die angeforderte Breite des inneren Randes dem Window zurück.
- j_getlayoutid** *int j_getlayoutid (int obj);*
Liefert die Eventnummer des Layoutmanagers des Containers **obj** zurück.
- j_getparentid** *int j_getparentid (int obj);*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getparent** *int j_getparent (int obj);*
Liefert die Eventnummer des umgebenen Containers von **obj** zurück. Ist **obj** ein Frame wird -1 zurückgegeben.
- j_getwidth** *int j_getwidth (int obj);*
Liefert die Breite des Window **obj** zurück.
- j_getxpos** *int j_getxpos (int obj);*
Liefert die aktuelle horizontale Position des Window **obj** zurück.
- j_getypos** *int j_getypos (int obj);*
Liefert die aktuelle vertikale Position des Window **obj** zurück.
- j_graphicbutton** *int j_graphicbutton (int obj , char* filename);*
Erzeugt einen neuen Graphicbutton mit dem Bildinhalt des Files, das durch **filename** spezifiziert wird.
- j_graphiclabel** *int j_graphiclabel (int obj , char* str);*
Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch **filename** spezifiziert wird.
- j_hide** *void j_hide (int obj);*
Zeigt das Window **obj** nicht mehr auf dem Bildschirm an.
- j_hscrollbar** *int j_hscrollbar (int obj);*
Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
- j_isparent** *int j_isparent (int obj , int cont);*
Liefert J_TRUE zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst J_FALSE
- j_isvisible** *int j_isvisible (int obj);*
Liefert J_TRUE zurück, falls **obj** sichtbar ist, sonst J_FALSE
- j_keylistener** *int j_keylistener (int obj);*
Bindet an das Window **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.

j_label	<i>int j_label (int obj , char* label);</i> Erzeugt ein neues Label Element mit dem Textinhalt label
j_led	<i>int j_led (int obj , int style , int color);</i> Erzeugt eine LED der Farbe color und der Form style
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
j_list	<i>int j_list (int obj , int rows);</i> Erzeugt eine neue List und gibt deren Eventnummer zurück.
j_meter	<i>int j_meter (int obj , char* title);</i> Erzeugt ein analoges Anzeigeelement mit der Beschriftung title .
j_mouselistener	<i>int j_mouselistener (int obj , int 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>void j_pack (int obj);</i> Veranlaßt den Layoutmanager von obj die minimale Größe zu berechnen.
j_panel	<i>int j_panel (int obj);</i> Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>void j_print (int obj);</i> Druckt den Inhalt des Window .
j_progressbar	<i>int j_progressbar (int obj , int orient);</i> Erzeugt eine Progress Bar mit der Ausrichtung orient .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück.
j_releaseall	<i>void j_releaseall (int obj);</i> Gibt alle Elemente aus dem Window obj frei.
j_release	<i>void j_release (int obj);</i> Gibt das Window obj aus dem umgebenden Container Element frei.
j_scrollpane	<i>int j_scrollpane (int obj);</i> Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.
j_setalign	<i>void j_setalign (int obj , int align);</i> Setzt die Ausrichtung innerhalb des Window obj auf align . Ein Flowlayout Manager wird vorausgesetzt.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i>

Integriert einen BorderLayout Manager in das Window **obj**.

j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Positioniert das Window obj an die angegebene Position pos . Im umgebenden Container muß einen BorderLayout Manager eingestellt sein.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Setzt den Cursor des Window obj auf cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Integriert einen Fixlayout Manager in das Window obj (voreingestellte Layoutmanager).
j_setflowfill	<i>void j_setflowfill (int obj , int 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>void j_setflowlayout (int obj , int align);</i> Integriert einen Flowlayout Manager in das Window obj mit der Ausrichtung align .
j_setfocus	<i>int j_setfocus (int obj);</i> Weist den Input Focus dem Window obj zu.
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Setzt den Fonttyp des Window obj auf name .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Setzt den Font des Window obj entsprechend der Parameterliste.
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Setzt den Fontgröße des Window obj auf size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Setzt den Fontstyle des Window obj auf style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Integriert einen Gridlayout Manager in das Window obj mit row Reihen und col Spalten.
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Setzt den horizontalen Abstand innerhalb des Window obj auf hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Setzt den inneren Rahmen auf die angegebenen Größen.

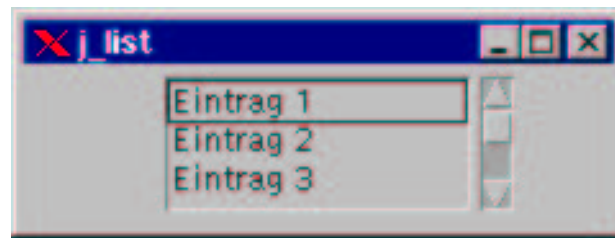
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Setzt die Hintergrundfarbe auf eine vordefinierte Farbe color .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Setzt die Vordergrundfarbe auf eine vordefinierte Farbe color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Entfernt den aktuellen Layout Manager aus dem Window obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Verschiebt das Window obj zur angegebenen Koordinate (xpos,ypos) an.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Setzt Breite und Höhe des Window obj auf width and height .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Setzt den vertikalen Abstand innerhalb des Window obj auf vgap Pixel.
j_sevenssegment	<i>int j_sevenssegment (int obj , int color);</i> Erzeugt eine Siebensegment Anzeige der Farbe color .
j_show	<i>void j_show (int obj);</i> Zeigt das Window obj auf dem Bildschirm an.
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>int j_windowlistener (int window , int 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	void j_additem (int obj , char* str);	
Argumente	obj str	int char*
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 <i>j_getselect()</i> ermittelt werden.	
Zielkomponenten	List, Choice	
Beispiel	<pre>: list = j_list(frame,3); j_additem(list,"Eintrag 1"); j_additem(list,"Eintrag 2"); :</pre>	



add

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

alertbox

Synopsis `void j_alertbox (int obj , char* title , char* text , char* button);`

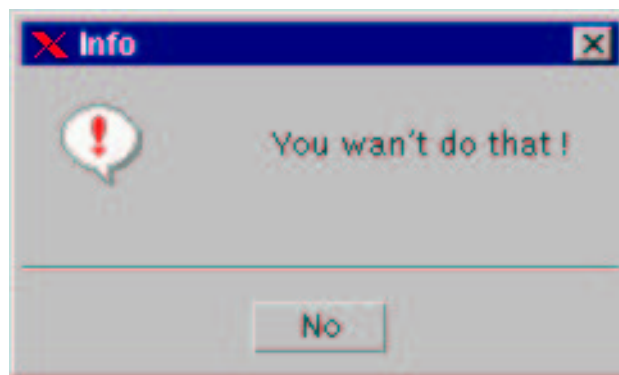
Argumente obj int
 title char*
 text char*
 button char*

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 `void j_appendtext (int obj , char* text);`

Argumente `obj` `int`
 `text` `char*`

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

Zielkomponenten Textarea

beep

Synopsis

```
void j_beep ( );
```

Beschreibung

Es ertönt ein Systembeep.

borderpanel

Synopsis `int j_borderpanel (int obj , int type);`

Argumente `obj int`
 `type int`

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.

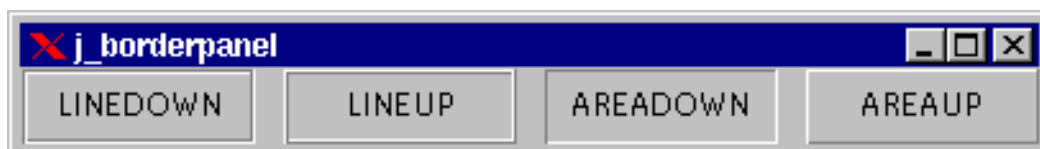
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
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 `int j_button (int obj , char* label);`

Argumente `obj` `int`
 `label` `char*`

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 `int j_canvas (int obj , int width , int height);`

Argumente `obj int`
 `width int`
 `height int`

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);
j_setnamedcolorbg(canvas,J_RED);
:
```



checkbox

Synopsis `int j_checkbox (int obj , char* label);`

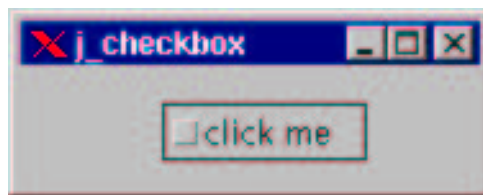
Argumente obj int
 label char*

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");  
:
```



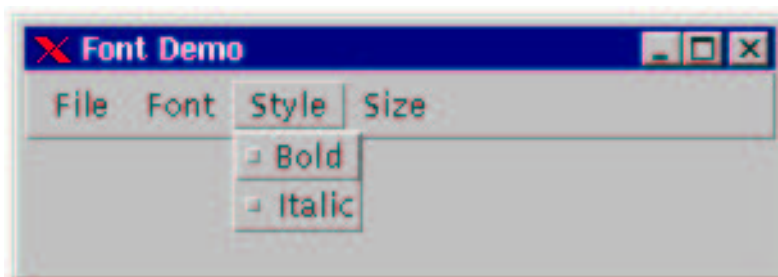
checkmenuitem

Synopsis	<code>int j_checkmenuitem (int obj , char* label);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj label</div> <div>int char*</div> </div>
Beschreibung	<p>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 <i>j_getstate()</i> abgefragt werden. Als Zielkomponenten sind nur Menu, Popupmenu und Helpmenu zulässig.</p>
Zielkomponenten	Menu, Popupmenu, Helpmenu
Beispiel	

```

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

```



choicebox2

Synopsis `void j_choicebox2 (int obj , char* title , char* text , char*
button1 , char* button2);`

Argumente

<code>obj</code>	<code>int</code>
<code>title</code>	<code>char*</code>
<code>text</code>	<code>char*</code>
<code>button1</code>	<code>char*</code>
<code>button2</code>	<code>char*</code>

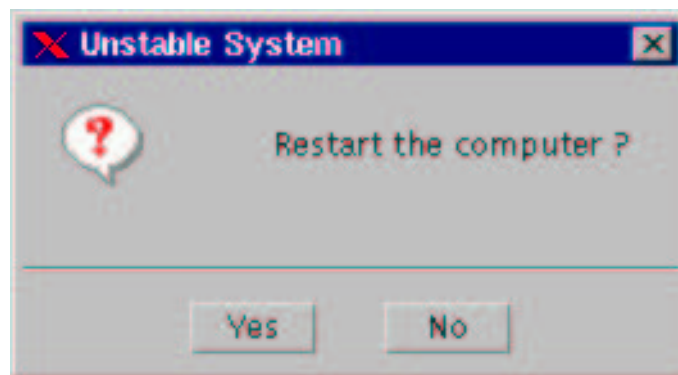
Beschreibung

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

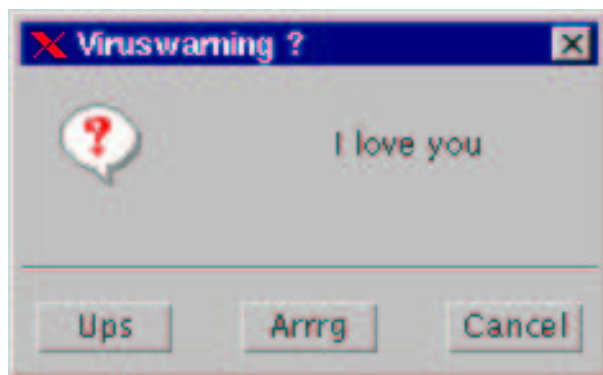
Beispiel

```
:  
retval = j_choicebox2(frame,"Unstable System","Restart the computer ?",  
                        "  Yes  ","No");  
:
```



checkboxbox3

Synopsis	void j_checkbox3 (int obj , char* title , char* text , char* button1 , char* button2 , char* button3);	
Argumente	obj title text button1 button2 button3	int char* char* char* char* char*
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	<pre>: retval = j_checkbox2(frame,"Viruswarning ?", "I love you", "Ups", "Arrrg", "Cancel"); :</pre>	



choice

Synopsis `int j_choice (int obj);`

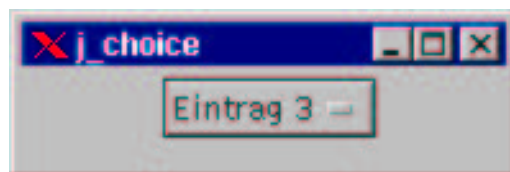
Argumente `obj` `int`

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);  
j_additem(choice,"Eintrag 1");  
j_additem(choice,"Eintrag 2");  
:
```



cliprect

Synopsis `void j_cliprect (int obj , int x , int y , int width , int height);`

Argumente `obj` `int`
 `x` `int`
 `y` `int`
 `width` `int`
 `height` `int`

Beschreibung Setzt den Clipping Bereich auf das spezifizierte Rechteck (**x**, **y**, **width**, **height**).

Zielkomponenten Canvas, Image, Printer

componentlistener

Synopsis	int j_componentlistener (int obj , int kind);	
Argumente	obj	int
	kind	int
Beschreibung	<p>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:</p> <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, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment	

connect

Synopsis	<code>int j_connect (char* hostname);</code>
Argumente	<code>hostname char*</code>
Beschreibung	<p>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 J_FALSE zurückgegeben</p>
Beispiel	<pre>: if(! j_connect("atan.japi.de")) or if(! j_connect("127.0.0.1")) :</pre>

delete

Synopsis	void j_delete (int obj , int start , int end);	
Argumente	obj	int
	start	int
	end	int
Beschreibung	löscht den Text von Position start bis Position end .	
Zielkomponenten	Textarea	

deselect

Synopsis	int j_deselect (int obj , int item);	
Argumente	obj	int
	item	int
Beschreibung	Deselectiert das Item mit dem Index item .	
Zielkomponenten	List	

dialog

Synopsis `int j_dialog (int obj , char* label);`

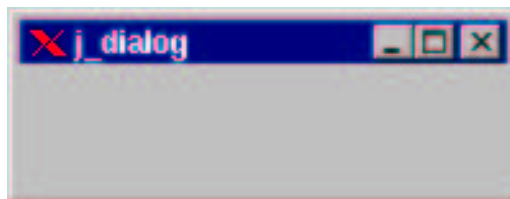
Argumente `obj` `int`
 `label` `char*`

Beschreibung Dies Funktion erzeugt ein neues Dialogwindow mit dem Titel **la-
 bel** und liefert dessen Eventnummer zurück. Im Fehlerfall wird
 -1 zurückgeliefert. Ein Dialogfenster ist ein eigenständiges Fen-
 ster, ebenso wie ein Frame. Allerdings kann ein Dialogfenster keine
 Menüleiste besitzen. Als Zielobjekt dieser Funktion ist nur ein Fra-
 me zulässig. Ein Dialog wird erst angezeigt, wenn ein entsprechen-
 des *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");
j_setsize(dialog,200,80);
j_show(dialog);
:
```



disable

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

dispose

Synopsis	<code>void j_dispose (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
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, Sevensegment, Canvas, Image, Printer, Keylistener, Focuslistener, Moucelistener

drawarc

Synopsis `void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);`

Argumente

<code>obj</code>	<code>int</code>
<code>x</code>	<code>int</code>
<code>y</code>	<code>int</code>
<code>rx</code>	<code>int</code>
<code>ry</code>	<code>int</code>
<code>arc1</code>	<code>int</code>
<code>arc2</code>	<code>int</code>

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.

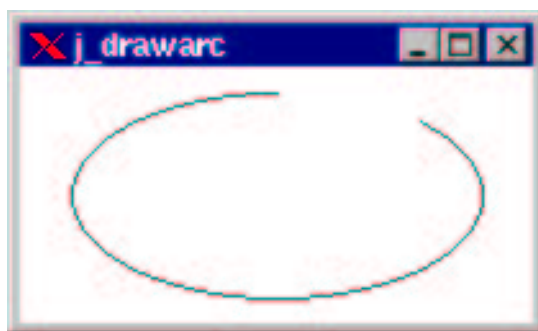
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



drawcircle

Synopsis `void j_drawcircle (int obj , int x , int y , int r);`

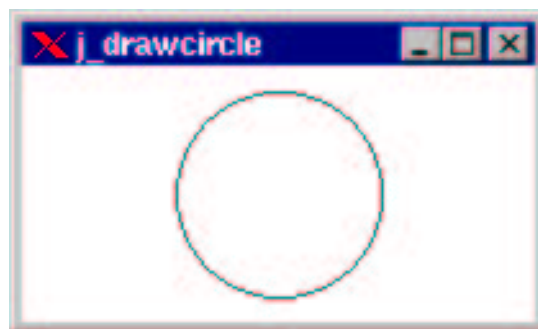
Argumente	<code>obj</code>	<code>int</code>
	<code>x</code>	<code>int</code>
	<code>y</code>	<code>int</code>
	<code>r</code>	<code>int</code>

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);  
j_drawcircle(canvas,100,50,40);  
:
```



drawimagesource

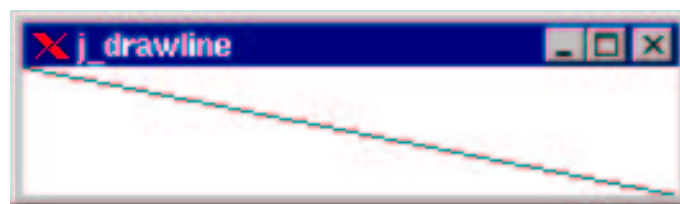
Synopsis	<code>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</code>	
Argumente	obj	int
	x	int
	y	int
	w	int
	h	int
	r	int*
	g	int*
	b	int*
Beschreibung	Die Funktion stellt ein Bild an der Position (x,y) mit der Breite w und der Höhe h dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays r,g,b übergeben.	
Zielkomponenten	Canvas, Image, Printer	

drawimage

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

drawline

Synopsis	<code>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</code>	
Argumente	obj	int
	x1	int
	y1	int
	x2	int
	y2	int
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	<pre> : canvas = j_canvas(frame,256,50); j_drawline(canvas,0,0,256,50); : </pre>	



drawoval

Synopsis `void j_drawoval (int obj , int x , int y , int rx , int ry);`

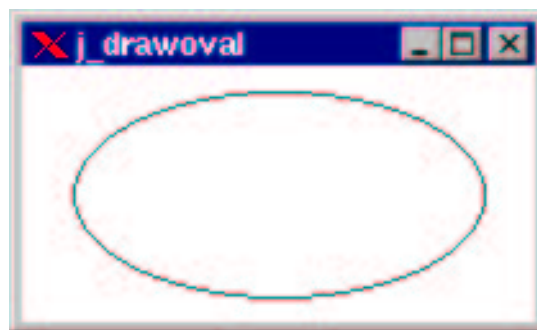
Argumente `obj` `int`
 `x` `int`
 `y` `int`
 `rx` `int`
 `ry` `int`

Beschreibung Zeichnet einen Oval mit dem Mittelpunkt (**x,y**) und den Hauptachsen **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

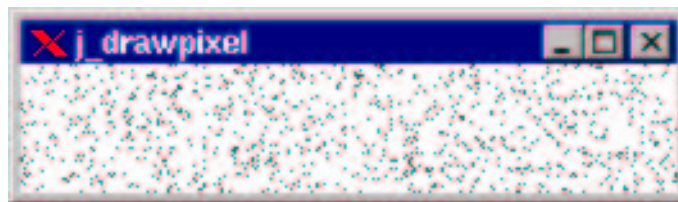
Beispiel

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



drawpixel

Synopsis	<code>void j_drawpixel (int obj , int x , int y);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>x</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>y</div> <div>int</div> </div>
Beschreibung	Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position (x,y). Bezugspunkt ist die linke obere Ecke der Komponenteneten, die durch obj referenziert wird.
Zielkomponenten	Canvas, Image, Printer
Beispiel	<pre> : canvas = j_canvas(frame,256,50); for(i=0;i<1000;i++) j_drawpixel(canvas,j_random()%256,,j_random()%256); : </pre>



drawpolygon

Synopsis `void j_drawpolygon (int obj , int len , int* x , int* y);`

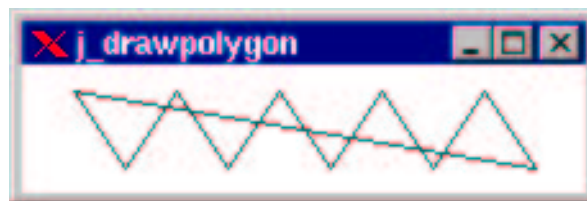
Argumente obj int
 len int
 x int*
 y int*

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

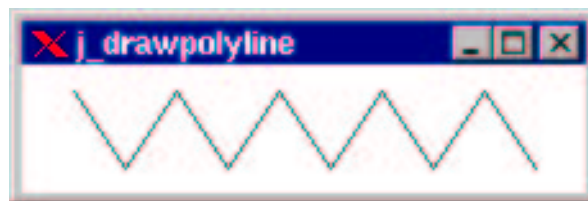
Beispiel

```
:
int x[10]={20,40,60,80,100,120,140,160,180,200};
int y[10]={10,40,10,40,10,40,10,40,10,40};
canvas = j_canvas(frame,256,50);
j_drawpolygon(canvas,10,x,y);
:
```



drawpolyline

Synopsis	<code>void j_drawpolyline (int obj , int len , int* x , int* y);</code>	
Argumente	obj	int
	len	int
	x	int*
	y	int*
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	<pre> : int x[10]={20,40,60,80,100,120,140,160,180,200}; int y[10]={10,40,10,40,10,40,10,40,10,40}; canvas = j_canvas(frame,256,50); j_drawpolyline(canvas,10,x,y); : </pre>	



drawrect

Synopsis `void j_drawrect (int obj , int x , int y , int width , int height);`

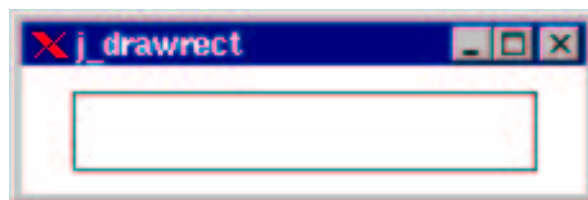
Argumente obj int
 x int
 y int
 width int
 height int

Beschreibung Zeichnet eine 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);  
j_drawrect(canvas,20,10,180,30);  
:
```



drawroundrect

Synopsis `void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);`

Argumente

<code>obj</code>	<code>int</code>
<code>x</code>	<code>int</code>
<code>y</code>	<code>int</code>
<code>width</code>	<code>int</code>
<code>height</code>	<code>int</code>
<code>arcx</code>	<code>int</code>
<code>arcy</code>	<code>int</code>

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.

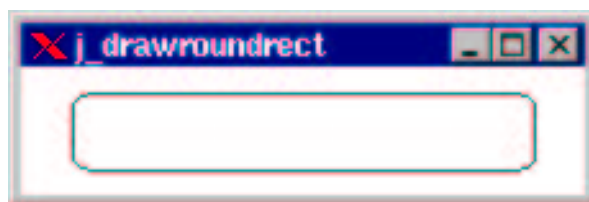
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



drawscaledimage

Synopsis	void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);	
Argumente	obj	int
	image	int
	sx	int
	sy	int
	sw	int
	sh	int
	tx	int
	ty	int
	tw	int
	th	int
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 `void j_drawstring (int obj , int x , int y , char* str);`

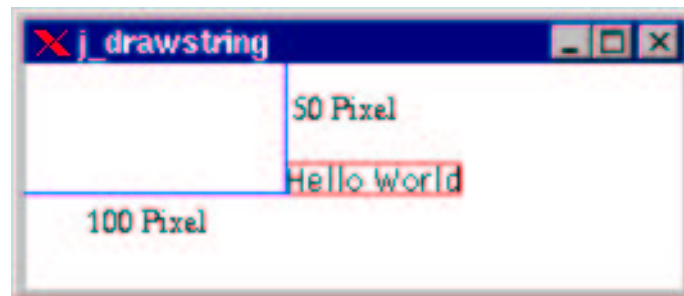
Argumente obj int
 x int
 y int
 str char*

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 gedank-
 lich um den gezeichneten String legen läßt.

Zielkomponenten Canvas, Image, Printer

Beispiel

```
:
j_drawstring(canvas,100,50,"Hello World");
:
```



enable

Synopsis `void j_enable (int obj);`

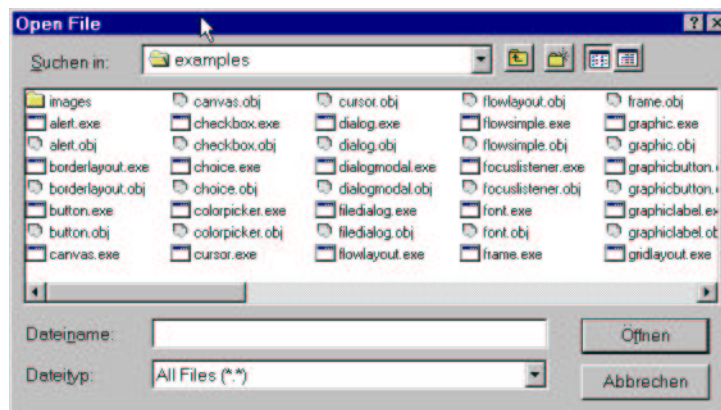
Argumente `obj` `int`

Beschreibung enabled die Komponente **obj**.

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

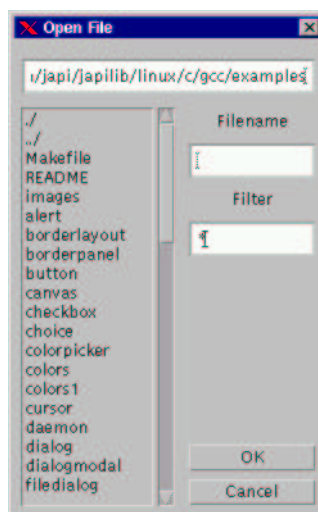
filedialog

Synopsis	<code>char* j_filedialog (int frame , char* title , char* directory , char* filename);</code>								
Argumente	<table> <tr><td>frame</td><td>int</td></tr> <tr><td>title</td><td>char*</td></tr> <tr><td>directory</td><td>char*</td></tr> <tr><td>filename</td><td>char*</td></tr> </table>	frame	int	title	char*	directory	char*	filename	char*
frame	int								
title	char*								
directory	char*								
filename	char*								
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	<pre> : filename = j_filedialog(frame,"Save/S File","..",filename); : </pre>								



fileselector

Synopsis	char* j_fileselector (int frame , char* title , char* filter , char* filename);	
Argumente	frame	int
	title	char*
	filter	char*
	filename	char*
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	<pre>: filename = j_fileselect(frame,"Open File","*",filename); :</pre>	



fillarc

Synopsis `void j_fillarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);`

Argumente

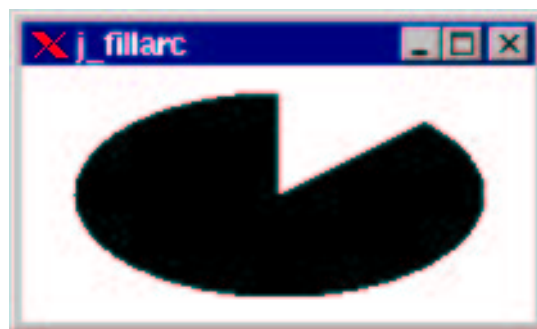
<code>obj</code>	<code>int</code>
<code>x</code>	<code>int</code>
<code>y</code>	<code>int</code>
<code>rx</code>	<code>int</code>
<code>ry</code>	<code>int</code>
<code>arc1</code>	<code>int</code>
<code>arc2</code>	<code>int</code>

Beschreibung 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. 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);
j_fillarc(canvas,100,50,80,40,45,-270);
:
```



fillcircle

Synopsis `void j_fillcircle (int obj , int x , int y , int r);`

Argumente `obj` `int`
 `x` `int`
 `y` `int`
 `r` `int`

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

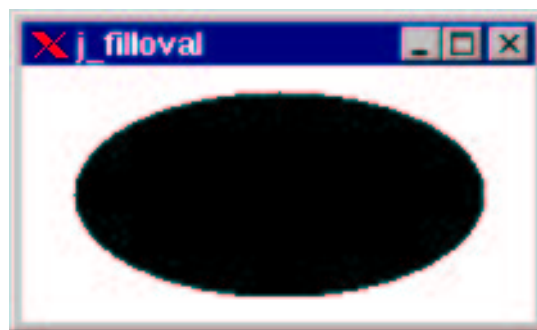
Beispiel

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



filloval

Synopsis	<code>void j_filloval (int obj , int x , int y , int rx , int ry);</code>	
Argumente	obj	int
	x	int
	y	int
	rx	int
	ry	int
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	<pre> : canvas = j_canvas(frame,200,100); j_filloval(canvas,100,50,80,40); : </pre>	



fillpolygon

Synopsis `void j_fillpolygon (int obj , int len , int* x , int* y);`

Argumente `obj` `int`
 `len` `int`
 `x` `int*`
 `y` `int*`

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

```
:
int x[10]={20,40,60,80,100,120,140,160,180,200};
int y[10]={10,40,10,40, 10,40,10,40,10,40};
canvas = j_canvas(frame,256,50);
j_fillpolygon(canvas,10,x,y);
:
```



fillrect

Synopsis `void j_fillrect (int obj , int x , int y , int width , int height);`

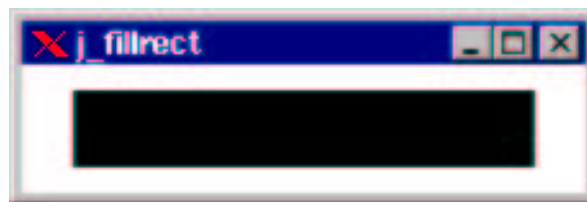
Argumente obj int
 x int
 y int
 width int
 height int

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);
j_fillrect(canvas,20,10,180,30);
:
```



fillroundrect

Synopsis `void j_fillroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);`

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

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.

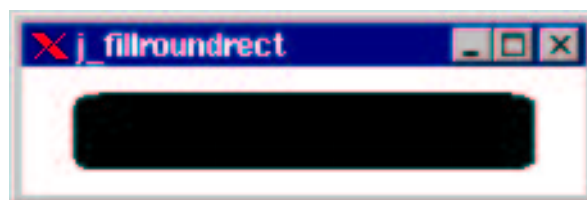
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



focuslistener

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

frame

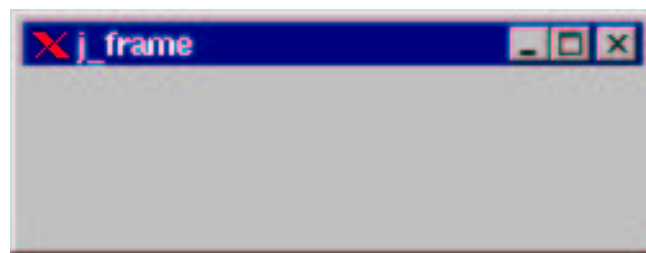
Synopsis `int j_frame (char* label);`

Argumente label char*

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");  
j_show(frame);  
:
```



getaction

Synopsis

int **j_getaction** ();

Beschreibung

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

getcolumns

Synopsis `void j_getcolumns (int obj);`

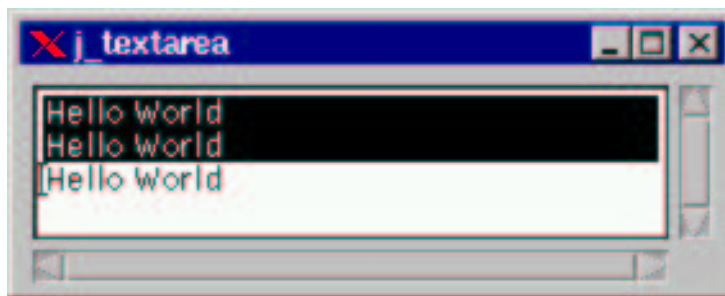
Argumente `obj` `int`

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

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

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



getcurpos

Synopsis	<code>int j_getcurpos (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Liefert die Position des Text Cursors zurück.
Zielkomponenten	Textarea, Textfield

getfontascent

Synopsis	int j_getfontascent (int obj);
Argumente	obj int
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, Sevensegment

getfontheight

Synopsis	int j_getfontheight (int obj);
Argumente	obj int
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, Sevensegment

getheight

Synopsis	int j_getheight (int obj);
Argumente	obj int
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"); printf("%s",j_getheight(label)); : > 22</pre>

getimagesource

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

getimage

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

getinsets

Synopsis `int j_getinsets (int obj , int side);`

Argumente obj int
 side int

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

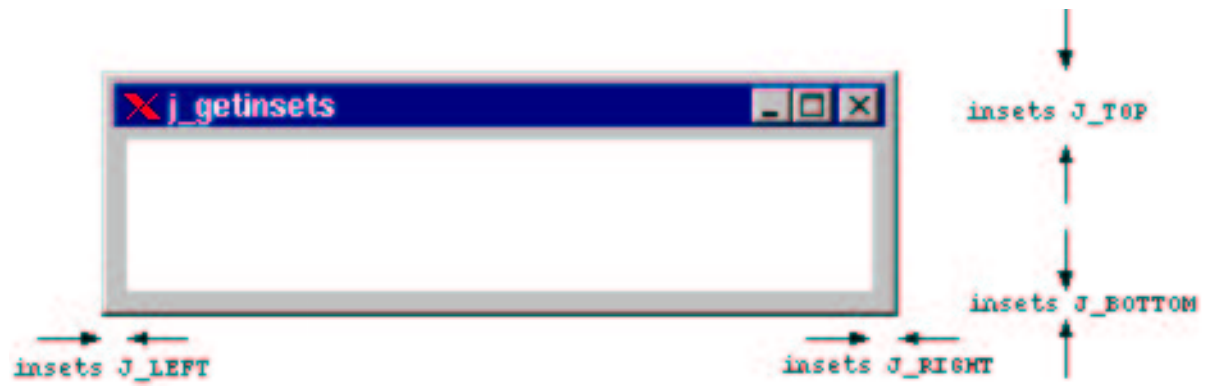
Beispiel

```

:
frame = j_frame("j_getinsets");
printf("%d %d %d %d\n",j_getinsets(frame,J_TOP),j_getinsets(frame,J_BOTTOM),
                                j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT));
:

> 25 5 5 6

```



getitemcount

Synopsis	int j_getitemcount (int obj);
Argumente	obj int
Beschreibung	Liefert die Anzahl der Items der Komponenten obj zurück.
Zielkomponenten	List, Choice

getitem

Synopsis	<code>char* j_getitem (int obj , int item , char* str);</code>	
Argumente	<code>obj</code>	<code>int</code>
	<code>item</code>	<code>int</code>
	<code>str</code>	<code>char*</code>
Beschreibung	liefert den Inhalt des Items mit dem Index item zurück.	
Zielkomponenten	List, Choice	

getkeychar

Synopsis	<code>int j_getkeychar (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.
Zielkomponenten	KeyListener

getkeycode

Synopsis	<code>int j_getkeycode (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Liefert den Keycode, der zuletzt gedrückten Taste zurück.
Zielkomponenten	KeyListener

getlayoutid

Synopsis	int j_getlayoutid (int obj);
Argumente	obj int
Beschreibung	Liefert die Eventnummer des Layoutmanagers des Containers obj zurück.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre>: j_setgridlayout(frame,2,2); grid = j_getlayoutid(frame); :</pre>

getlength

Synopsis	<code>int j_getlength (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Liefert die Länge des Textinhaltes die Komponente zurück.
Zielkomponenten	Textarea, Textfield, Dialog, Frame, Button, MenuItem, CheckBox- MenuItem, Menu, HelpMenu, Popupmenu

getmousebutton

Synopsis	int j_getmousebutton (int mouset listener);
Argumente	mouset listeneint
Beschreibung	<p>Liefert die letzte gedrückte Moustaste zurück. Der Rückgabewert ist</p> <ul style="list-style-type: none">• J_LEFT linke Moustaste• J_CENTER mittlere Moustaste• J_RIGHT rechte Moustaste
Zielkomponenten	Mouset listener

getmousex

Synopsis	int j_getmousex (int mouset listener);
Argumente	mouset listenerint
Beschreibung	Liefert die aktuelle horizontale Position der Mouse zurück.
Zielkomponenten	Mouset listener

getmousey

Synopsis	int j_getmousey (int mouset listener);
Argumente	mouset listenerint
Beschreibung	Liefert die aktuelle vertikale Position der Mouse zurück.
Zielkomponenten	Mouset listener

getparentid

Synopsis `int j_getparentid (int obj);`

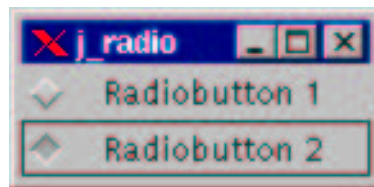
Argumente `obj` `int`

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, Sevensegment, 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 `int j_getparent (int obj);`

Argumente `obj` `int`

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

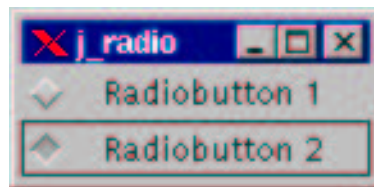
Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Beispiel

```

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

```



getrows

Synopsis `void j_getrows (int obj);`

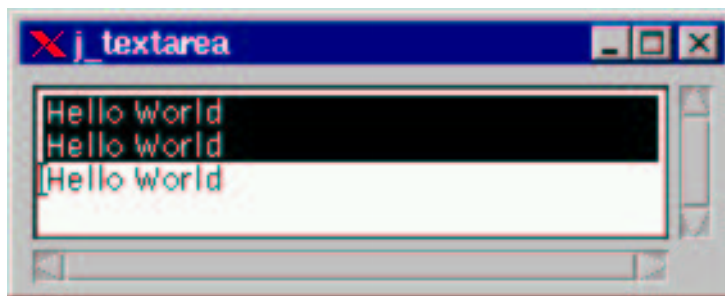
Argumente `obj` `int`

Beschreibung Gibt die Anzahl der Reihen von **obj** zurck.

Zielkomponenten Textarea, Gridlayout

Beispiel

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



getscaledimage

Synopsis	int j_getscaledimage (int obj , int x , int y , int sw , int sh , int tw , int th);	
Argumente	obj	int
	x	int
	y	int
	sw	int
	sh	int
	tw	int
	th	int
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

```
int j_getscreenheight ( );
```

Beschreibung

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

Beispiel

```
:  
printf("%d %d\n", j_getscreenwidth(), j_getscreenheight());  
:  
  
> 1280 1024
```


getscreenwidth

Synopsis

```
int j_getscreenwidth ( );
```

Beschreibung

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

Beispiel

```
:  
printf("%d %d\n", j_getscreenwidth(), j_getscreenheight());  
:  
  
> 1280 1024
```

getselect

Synopsis `int j_getselect (int obj);`

Argumente `obj` `int`

Beschreibung Liefert den Index des selectierten Items der Komponenten .

Zielkomponenten `List, Choice`

getselend

Synopsis `int j_getselend (int obj);`

Argumente `obj` `int`

Beschreibung Liefert die Endposition des selectierten Textes.

Zielkomponenten Textarea, Textfield

getselstart

Synopsis	<code>int j_getselstart (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Liefert die Anfangsposition des selectierten Textes.
Zielkomponenten	Textarea, Textfield

getseltext

Synopsis	<code>char* j_getseltext (int obj , char* text);</code>	
Argumente	<code>obj</code>	<code>int</code>
	<code>text</code>	<code>char*</code>
Beschreibung	Liefert den selectierten Text die Komponente obj zurück.	
Zielkomponenten	Textarea, Textfield	

getstate

Synopsis	<code>int j_getstate (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Liefert <code>J_TRUE</code> , falls die Komponente selektiert ist, sonst <code>J_FALSE</code> .
Zielkomponenten	Checkbox, Radiobutton, Checkmenuitem, Led

gettext

Synopsis	<code>char* j_gettext (int obj , char* str);</code>				
Argumente	<table> <tr> <td><code>obj</code></td><td><code>int</code></td></tr> <tr> <td><code>str</code></td><td><code>char*</code></td></tr> </table>	<code>obj</code>	<code>int</code>	<code>str</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>str</code>	<code>char*</code>				
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>char str[256]; : label = j_label(frame,"Hello World"); printf("%s",j_gettext(label,str)); : > Hello World</pre>				

getvalue

Synopsis `int j_getvalue (int obj);`

Argumente `obj` `int`

Beschreibung Liefert den aktuellen Wert der Komponenten .

Zielkomponenten Scrollbar

getviewportheight

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

getviewportwidth

Synopsis	int j_getviewportwidth (int obj);
Argumente	obj int
Beschreibung	Gibt die Breite des sichtbaren Teils der Komponenten obj zurück.
Zielkomponenten	Scrollpane

getwidth

Synopsis	int j_getwidth (int obj);
Argumente	obj int
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"); printf("%s",j_getwidth(label)); : </pre> <p>> 84</p>

getxpos

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

graphicbutton

Synopsis `int j_graphicbutton (int obj , char* filename);`

Argumente `obj` `int`
 `filename` `char*`

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 `int j_graphiclabel (int obj , char* str);`

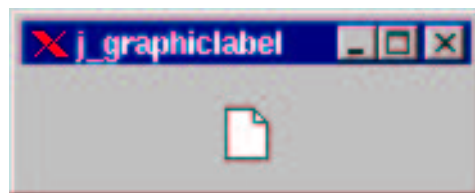
Argumente `obj` `int`
 `str` `char*`

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	int j_hasfocus (int obj);
Argumente	obj int
Beschreibung	Liefert J_TRUE zurück, falls die Komponente den Focus besitzt, sonst J_FALSE .
Zielkomponenten	Focuslistener

helpmenu

Synopsis `int j_helpmenu (int obj , char* label);`

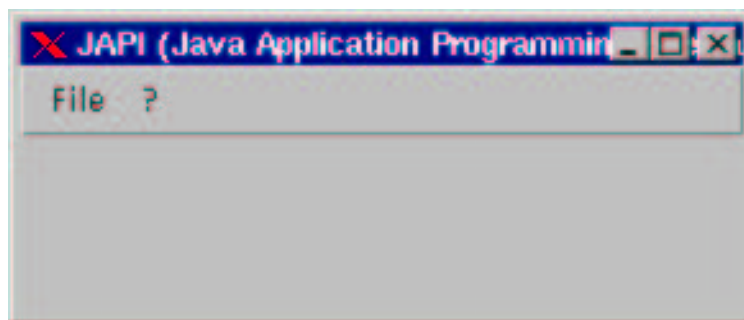
Argumente `obj` `int`
 `label` `char*`

Beschreibung Die Funktion erzeugt einen Hilfe-Menueintrag mit der Beschriftung **label** in der Komponenten **obj**, und liefert dessen Eventnummer zurück. An diese Menukomponente können in weiteren Schritten MenuItem gebunden werden. Eine Menukomponente selbst erzeugt keinen Event. Hilfe Menü unterscheiden sich von normalen Menüs dadurch, daß sie je nach Oberfläche auch rechtebündig 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	<code>void j_hide (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
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 `int j_hscrollbar (int obj);`

Argumente `obj` `int`

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 ein Event, sobald der Schieber bewegt wird.

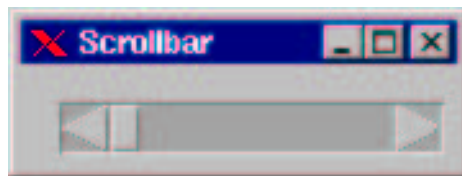
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame, Scrollpane

Beispiel

```

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

```



image

Synopsis `int j_image (int width , int height);`

Argumente width int
 height int

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	int j_insert (int obj , int pos , char* label);	
Argumente	obj	int
	pos	int
	label	char*
Beschreibung	fügt ein neues Item in an der Position pos ein. Das Item trägt den Inhalt label .	
Zielkomponenten	List, Choice	

inserttext

Synopsis `void j_inserttext (int obj , char* text , int pos);`

Argumente	obj	int
	text	char*
	pos	int

Beschreibung Fügt zusätzlichen Text **text** an der Position **pos** ein.

Zielkomponenten Textarea

isparent

Synopsis	int j_isparent (int obj , int cont);				
Argumente	<table> <tr> <td>obj</td><td>int</td></tr> <tr> <td>cont</td><td>int</td></tr> </table>	obj	int	cont	int
obj	int				
cont	int				
Beschreibung	Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE				
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup				

isresizable

Synopsis	int j_isresizable (int obj);
Argumente	obj int
Beschreibung	Liefert J_FALSE zurueck, falls die Komponente eine fixe Größe besitzt, J_TRUE sonst.
Zielkomponenten	Dialog, Frame

isselect

Synopsis	int j_isselect (int obj , int item);	
Argumente	obj	int
	item	int
Beschreibung	Liefert J_TRUE zurück, falls das Item mit dem Index item selektiert ist, sonst J_FALSE	
Zielkomponenten	List	

isvisible

Synopsis `int j_isvisible (int obj);`

Argumente `obj` `int`

Beschreibung Liefert J_TRUE zurück, falls **obj** sichtbar ist, sonst J_FALSE

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

label

Synopsis `int j_label (int obj , char* label);`

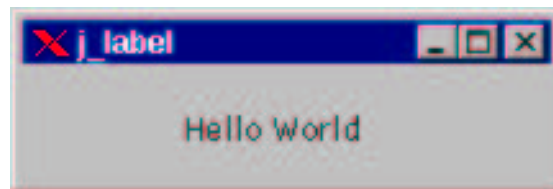
Argumente `obj` `int`
 `label` `char*`

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 `int j_led (int obj , int style , int color);`

Argumente `obj` `int`
 `style` `int`
 `color` `int`

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 `int j_line (int obj , int orient , int style , int length);`

Argumente	<code>obj</code>	<code>int</code>
	<code>orient</code>	<code>int</code>
	<code>style</code>	<code>int</code>
	<code>length</code>	<code>int</code>

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.

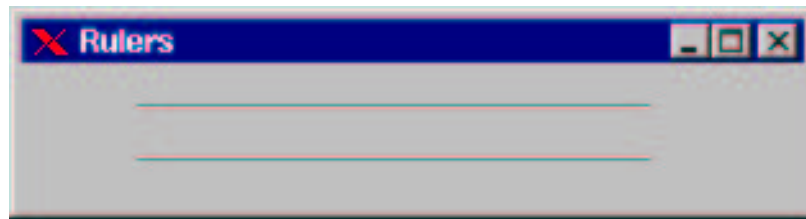
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

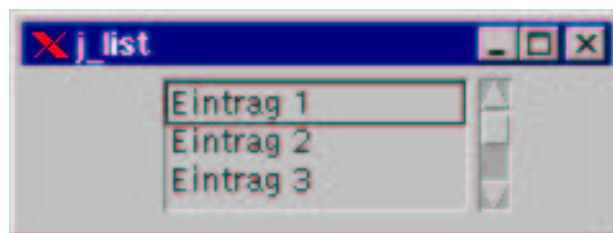
:
line1 = j_line(frame,J_HORIZONTAL,J_LINEDOWN,200);
line2 = j_line(frame,J_HORIZONTAL,J_LINEUP,200);
:

```



list

Synopsis	<code>int j_list (int obj , int rows);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>rows</div> <div>int</div> </div>
Beschreibung	<p>Diese Funktion erzeugt eine neue List und gibt deren Eventnummer zurück. Im Fehlerfall wird <code>-1</code> 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 <i><code>j_multiplemode()</code></i>). 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.</p>
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre> : list = j_list(frame,3); j_additem(list,"Eintrag 1"); j_additem(list,"Eintrag 2"); : </pre>



loadimage

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

menubar

Synopsis `int j_menubar (int obj);`

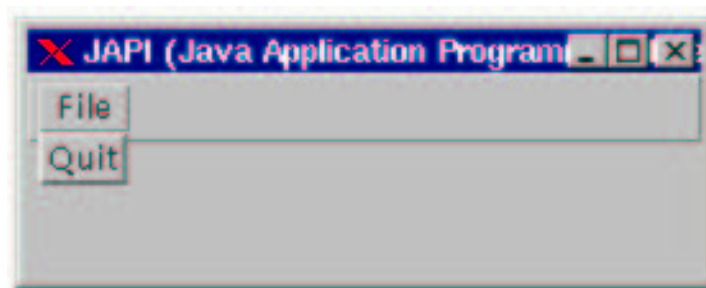
Argumente `obj` `int`

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 `int j_menuitem (int obj , char* label);`

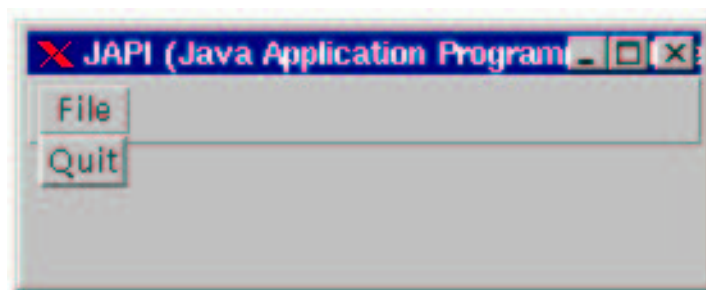
Argumente `obj` `int`
 `label` `char*`

Beschreibung Die Funktion erzeugt ein MenuItem mit der Beschriftung **label** in der Komponente **obj**, und liefert dessen Eventnummer zurück. Ein MenuItem erzeugt einen Event, wenn es durch einen Mausklick ausgewählt 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 `int j_menu (int obj , char* str);`

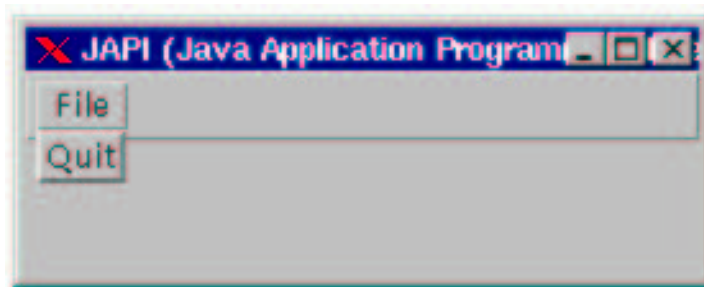
Argumente `obj` `int`
 `str` `char*`

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 `void j_messagebox (int obj , char* title , char* text);`

Argumente `obj int`
 `title char*`
 `text char*`

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 ein Event, wenn das Close Icon im Fensterrahmen angeklickt wird.

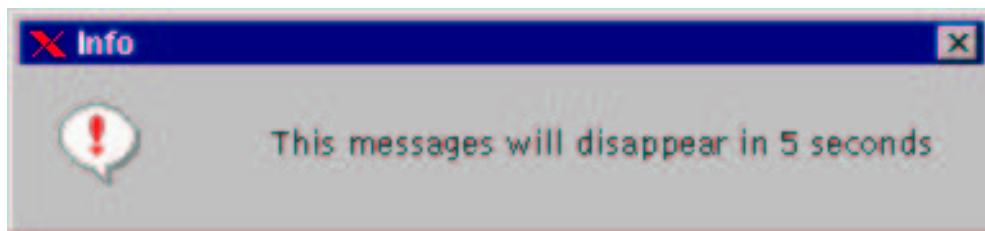
Zielkomponenten Frame

Beispiel

```

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

```



meter

Synopsis `int j_meter (int obj , char* title);`

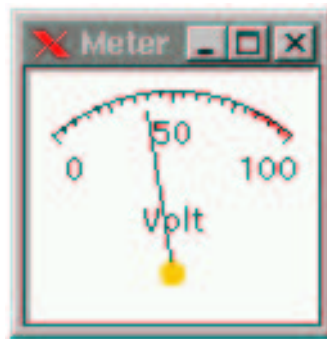
Argumente `obj` `int`
 `title` `char*`

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");
j_setvalue(meter,90);
:
```



mouselistener

Synopsis	<code>int j_mouselistener (int obj , int kind);</code>				
Argumente	<table> <tr> <td><code>obj</code></td><td><code>int</code></td></tr> <tr> <td><code>kind</code></td><td><code>int</code></td></tr> </table>	<code>obj</code>	<code>int</code>	<code>kind</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>kind</code>	<code>int</code>				
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 Moustaste 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 Moustaste 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	int j_multiplemode (int obj , int bool);		
Argumente	obj	int	
	bool	int	
Beschreibung	die Komponente erlaubt mehrfache Selectierung, falls bool J_TRUE ist.		
Zielkomponenten	List		

nextaction

Synopsis

int **j_nextaction** ();

Beschreibung

Wartet auf den nächsten Benutzerevent und liefert diesen zurück.

pack

Synopsis `void j_pack (int obj);`

Argumente `obj` `int`

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

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



panel

Synopsis `int j_panel (int obj);`

Argumente `obj` `int`

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.

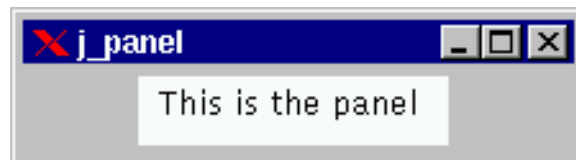
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



popupmenu

Synopsis `int j_popupmenu (int obj , char* label);`

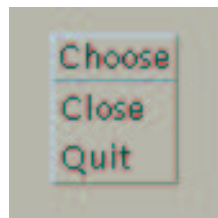
Argumente `obj` `int`
 `label` `char*`

Beschreibung Diese Funktion erzeugt ein neues Popupmenu und liefert dessen Eventnummer zurück. Im Fehlerfall wird `-1` zurückgegeben. Ein Popupmenu ist funktional identisch 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 Menuitems gebunden werden, die bei Anklicken des Popupmenus sichtbar werden.

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

Beispiel

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



printer

Synopsis	<code>int j_printer (int frame);</code>
Argumente	<code>frame</code> <code>int</code>
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 <code>-1</code> . Ein Printer erzeugt keinen Event.
Zielkomponenten	Frame
Beispiel	<pre>: printer = j_printer(frame); j_drawimage(printer,image,100,100); :</pre>

print

Synopsis `void j_print (int obj);`

Argumente `obj` `int`

Beschreibung 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.

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

Beispiel

```
:
frame = j_frame("j_textfield");
text  = j_textfield(frame,30)
:
j_print(frame);
:
```



progressbar

Synopsis **int j_progressbar** (int obj , int orient);

Argumente obj int
 orient int

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.

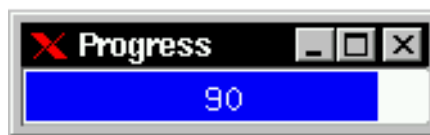
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



quit

Synopsis

```
void j_quit ( );
```

Beschreibung

Beendet die Verbindung zum JAPI Kernel.

radiobutton

Synopsis	<code>int j_radiobutton (int obj , char* label);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>label</div> <div>char*</div> </div>
Beschreibung	<p>Diese Funktion erzeugt einen Radiobutton mit dem Textinhalt label und liefert dessen Eventnummer zurück. Im Fehlerfall wird <code>-1</code> 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.</p>
Zielkomponenten	Radiogroup
Beispiel	<pre> : radiogroup = j_radiogroup(frame); radio1 = j_radiobutton(radiogroup,"Radiobutton 1"); radio2 = j_radiobutton(radiogroup,"Radiobutton 2"); : </pre>



radiogroup

Synopsis `int j_radiogroup (int obj);`

Argumente `obj` `int`

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

```
int j_random ( );
```

Beschreibung

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

releaseall

Synopsis	<code>void j_releaseall (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Gibt alle Elemente aus der Komponenten obj frei.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

release

Synopsis	<code>void j_release (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Gibt die Komponente obj aus dem umgebenden Container Element frei.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

removeall

Synopsis `int j_removeall (int obj);`

Argumente `obj` `int`

Beschreibung entfernt alle Items aus der Komponenten .

Zielkomponenten List, Choice

removeitem

Synopsis	int j_removeitem (int obj , char* item);	
Argumente	obj	int
	item	char*
Beschreibung	entfernt den ersten Eintrag item aus der Komponenten .	
Zielkomponenten	List, Choice	

remove

Synopsis `int j_remove (int obj , int item);`

Argumente	<code>obj</code>	<code>int</code>
	<code>item</code>	<code>int</code>

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

Zielkomponenten List, Choice

replacetext

Synopsis	<code>void j_replacetext (int obj , char* text , int start , int end);</code>	
Argumente	obj	int
	text	char*
	start	int
	end	int
Beschreibung	Ersetzt den Text zwischen Position start bis end durch text .	
Zielkomponenten	Textarea	

saveimage

Synopsis	<code>int j_saveimage (int obj , char* filename , int filetype);</code>						
Argumente	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>filename</code></td><td><code>char*</code></td></tr><tr><td><code>filetyp</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>filename</code>	<code>char*</code>	<code>filetyp</code>	<code>int</code>
<code>obj</code>	<code>int</code>						
<code>filename</code>	<code>char*</code>						
<code>filetyp</code>	<code>int</code>						
Beschreibung	<p>Speichert den Inhalt der Komponenten in einem File mit dem Namen filename. Es werden zwei Fileformate unterstützt:</p> <ul style="list-style-type: none">• BMP• PPM						
Beispiel	<pre>: if(! j_saveimage(canvas,"mandel.bmp",J_BMP)) printf("Error saving Bitmap file\n"); :</pre>						

scrollpane

Synopsis `int j_scrollpane (int obj);`

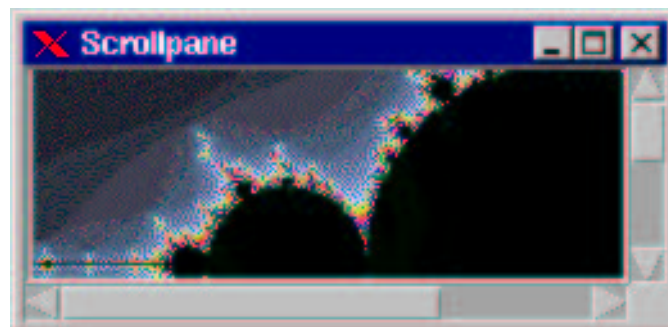
Argumente `obj` `int`

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");
j_setsize(scrollpane,240,100);
:
```



selectall

Synopsis `void j_selectall (int obj);`

Argumente `obj` `int`

Beschreibung Selectiert den gesamten Textinhalt der Komponenten .

Zielkomponenten Textarea, Textfield

select

Synopsis	int j_select (int obj , int item);	
Argumente	obj	int
	item	int
Beschreibung	selectiert das Item mit dem Index item .	
Zielkomponenten	List, Choice	

selecttext

Synopsis `void j_selecttext (int obj , int start , int end);`

Argumente	<code>obj</code>	<code>int</code>
	<code>start</code>	<code>int</code>
	<code>end</code>	<code>int</code>

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

Zielkomponenten Textarea, Textfield

seperator

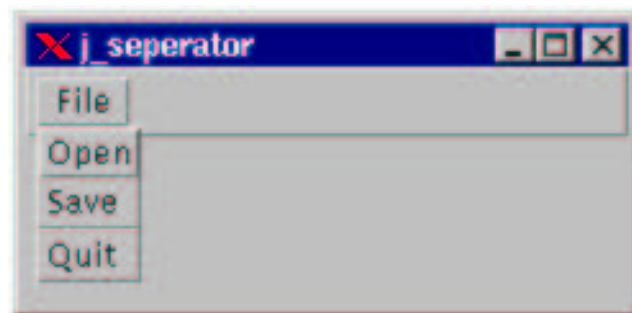
Synopsis	<code>void j_seperator (int obj);</code>
Argumente	obj int
Beschreibung	Diese Prozedur erzeugt eine Trennlinie in einem Menu, das durch obj referenziert wird. Trennlinien dienen ausschließlich der optischen Trennung von Menuitem, und besitzen keine weitere Funktionalität.
Zielkomponenten	Menu, HelpMenu, Popupmenu

Beispiel

```

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

```



setalign

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

setblockinc

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

setborderlayout


Synopsis	<code>void j_setborderlayout (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Integriert einen BorderLayout Manager in die Komponente obj .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setborderpos

Synopsis	void j_setborderpos (int obj , int pos);	
Argumente	obj	int
	pos	int
Beschreibung	Positioniert die Komponente obj an die angegebene Position pos . Im umgebenden Container muß einen 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	<code>void j_setcolorbg (int obj , int r , int g , int b);</code>								
Argumente	<table> <tr><td>obj</td><td>int</td></tr> <tr><td>r</td><td>int</td></tr> <tr><td>g,</td><td>int</td></tr> <tr><td>b</td><td>int</td></tr> </table>	obj	int	r	int	g,	int	b	int
obj	int								
r	int								
g,	int								
b	int								
Beschreibung	<p>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.</p>								
Zielkomponenten	<p>Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment</p>								
Beispiel	<pre> : button = j_button(frame,"Hello World"); j_setcolorbg(button,150,0,0); j_settext(button,"Hello World"); : </pre>								



setcolor

Synopsis	<code>void j_setcolor (int obj , int r , int g , int b);</code>
Argumente	<div style="display: flex; align-items: center;"> <div style="width: 100px;"></div> <div style="display: flex; flex-direction: column; gap: 5px;"> <div>obj int</div> <div>r int</div> <div>g, int</div> <div>b int</div> </div> </div>
Beschreibung	<p>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.</p>
Zielkomponenten	<p>Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment</p>
Beispiel	<pre> : button = j_button(frame,"Hello World"); j_setcolor(button,150,0,0); j_settext(button,"Hello World"); : </pre>



setcolumns

Synopsis `void j_setcolumns (int obj , int columns);`

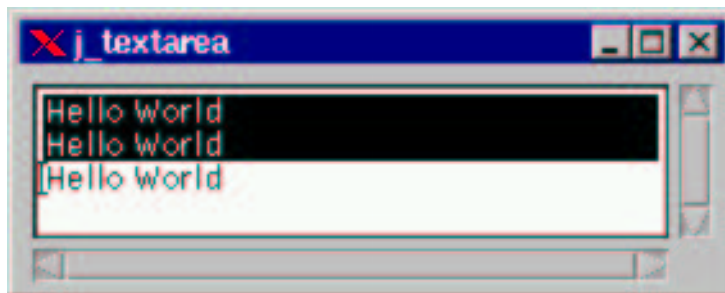
Argumente `obj` `int`
 `columns` `int`

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

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

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



setcurpos

Synopsis	void j_setcurpos (int obj , int pos);	
Argumente	obj	int
	pos	int
Beschreibung	Setzt den Text Cursor auf die Position pos .	
Zielkomponenten	Textarea, Textfield	

setcursor

Synopsis	int j_setcursor (int obj , int cursor);	
Argumente	obj	int
	cursor	int
Beschreibung	Setzt den Cursor der Komponenten obj auf cursor .	
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment	

setdebug

Synopsis	<code>void j_setdebug (int level);</code>
Argumente	<code>level</code> <code>int</code>
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. <i>j_setdebug</i> ist die einzige Funktion, die aufgerufen werden kann, bevor <i>j_start()</i> aufgerufen wurde. Die Levelebenen 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	void j_setechochar (int obj , char chr);	
Argumente	obj	int
	chr	char
Beschreibung	Bestimmt das Zeichen chr als Echo Zeichen für alle Benutzereingaben in die Komponente .	
Zielkomponenten	Textfield	

seteditable

Synopsis	void j_seteditable (int obj , int bool);	
Argumente	obj	int
	bool	int
Beschreibung	Ist bool J_TRUE , so ist die Komponente editierbar, ansonsten ist die Komponente nur lesbar für den Benutzer.	
Zielkomponenten	Textarea, Textfield	

setfixlayout

Synopsis	<code>void j_setfixlayout (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Integriert einen Fixlayout Manager in die Komponente obj (voreingestellte Layoutmanager).
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setflowfill

Synopsis	void j_setflowfill (int obj , int bool);	
Argumente	obj	int
	bool	int
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	void j_setflowlayout (int obj , int align);	
Argumente	obj	int
	align	int
Beschreibung	Integriert einen Flowlayout Manager in die Komponente obj mit der Ausrichtung align .	
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame	

setfocus

Synopsis	int j_setfocus (int obj);
Argumente	obj int
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

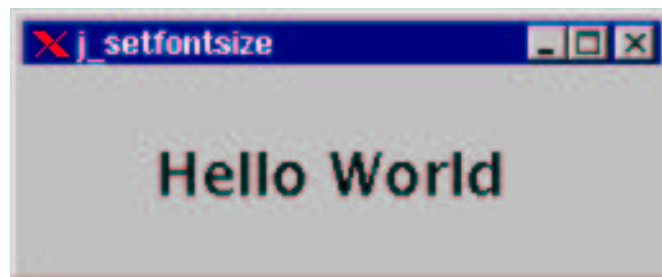
setfontname

Synopsis	<code>void j_setfontname (int obj , int name);</code>				
Argumente	<table> <tr> <td><code>obj</code></td><td><code>int</code></td></tr> <tr> <td><code>name</code></td><td><code>int</code></td></tr> </table>	<code>obj</code>	<code>int</code>	<code>name</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>name</code>	<code>int</code>				
Beschreibung	<p>Diese Prozedur setzt den Fonttyp in der Komponenten obj, auf name.</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 				
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				
Beispiel	<pre> : label = j_label(jframe,"Hello World"); j_setfontname(label,J_HELVETIA); : </pre>				



setfontsize

Synopsis	<code>void j_setfontsize (int obj , int size);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>size</div> <div>int</div> </div>
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, Popupmenu
Beispiel	<pre> : label = j_label(jframe,"Hello World"); j_setfontsize(label,24); : </pre>



setfontstyle

Synopsis	<code>void j_setfontstyle (int obj , int style);</code>				
Argumente	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>style</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>style</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>style</code>	<code>int</code>				
Beschreibung	<p>Diese Prozedur setzt den Fontstyle in der Komponenten, die durch obj referenziert wird, auf style. Für den Parameter style sind folgende Werte möglich:</p> <ul style="list-style-type: none">• <code>J_PLAIN</code> : normal• <code>J_BOLD</code> : fett• <code>J_ITALIC</code> : italic• <code>J_BOLD + J_ITALIC</code> : italic fett				
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				
Beispiel	<pre>: label = j_label(jframe,"Hello World"); j_setfontstyle(label,J_BOLD+J_ITALIC); :</pre>				



setfont

Synopsis	<code>void j_setfont (int obj , int name , int style , int size);</code>								
Argumente	<table> <tr> <td><code>obj</code></td><td><code>int</code></td></tr> <tr> <td><code>name</code></td><td><code>int</code></td></tr> <tr> <td><code>style</code></td><td><code>int</code></td></tr> <tr> <td><code>size</code></td><td><code>int</code></td></tr> </table>	<code>obj</code>	<code>int</code>	<code>name</code>	<code>int</code>	<code>style</code>	<code>int</code>	<code>size</code>	<code>int</code>
<code>obj</code>	<code>int</code>								
<code>name</code>	<code>int</code>								
<code>style</code>	<code>int</code>								
<code>size</code>	<code>int</code>								
Beschreibung	<p>Diese Prozedur setzt erzeugt einen neuen Font in der Komponente, 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, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu								

Beispiel

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



setgridlayout

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

sethgap

Synopsis `void j_sethgap (int obj , int hgap);`

Argumente `obj` `int`
 `hgap` `int`

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

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



setIcon

Synopsis	void j_seticon (int frame , int icon);	
Argumente	frame	int
	icon	int
Beschreibung	Ordnet dem frame das Icon icon zu. Eine iconisierte Darstellung wird nicht von allen Plattformen unterstützt.	
Zielkomponenten	Frame	
Beispiel	<pre>: frame = j_frame("Hello World"); j_seticon(frame,j_loadimage("icon.gif")); :</pre>	

setimage

Synopsis `void j_setimage (int obj , int image);`

Argumente `obj` `int`
 `image` `int`

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

Zielkomponenten Graphicbutton, Graphiclabel

Beispiel

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



setinsets

Synopsis `void j_setinsets (int obj , int top , int bottom , int left , int right);`

Argumente

<code>obj</code>	<code>int</code>
<code>top</code>	<code>int</code>
<code>bottom</code>	<code>int</code>
<code>left</code>	<code>int</code>
<code>right</code>	<code>int</code>

Beschreibung Setzt den inneren Rahmen auf die angegebenen Größen.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:
frame = j_frame("j_getinsets");
j_setinsets(frame,30,10,10,10);
:

```



setmax

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

setmin

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

setnamedcolorbg

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

setnamedcolor

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

setnolayout

Synopsis	<code>void j_setnolayout (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
Beschreibung	Entfernt den aktuellen Layout Manager aus der Komponenten obj .
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

setpos

Synopsis	void j_setpos (int obj , int xpos , int ypos);		
Argumente	obj	int	
	xpos	int	
	ypos	int	
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	<code>int j_setradiogroup (int rbutton, , int rgroup);</code>				
Argumente	<table><tr><td><code>rbutton,</code></td><td><code>int</code></td></tr><tr><td><code>rgroup</code></td><td><code>int</code></td></tr></table>	<code>rbutton,</code>	<code>int</code>	<code>rgroup</code>	<code>int</code>
<code>rbutton,</code>	<code>int</code>				
<code>rgroup</code>	<code>int</code>				
Beschreibung	Weist dem Radiobutton rbutton die Radiogroup rgroup zu. Der Radiobutton wird aus der alten Radiogroup entfernt.				
Zielkomponenten	Radiobutton				

setresizable

Synopsis	<code>void j_setresizable (int obj , int resizable);</code>				
Argumente	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>resizable</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>resizable</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>resizable</code>	<code>int</code>				
Beschreibung	Ist resizable <code>J_FALSE</code> , kann die Größe der Komponenten nicht vom Benutzer verändert werden.				
Zielkomponenten	Dialog, Frame				
Beispiel	<pre>: frame = j_frame("fixsized Frame"); j_setresizable(frame,J_FALSE); :</pre>				

setrows

Synopsis `void j_setrows (int obj , int rows);`

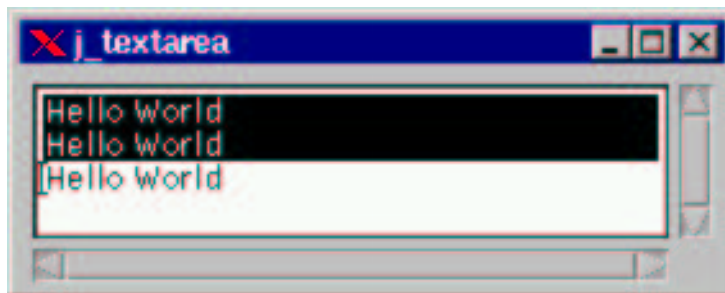
Argumente `obj` `int`
 `rows` `int`

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

Zielkomponenten Textarea, Gridlayout

Beispiel

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



setshortcut

Synopsis	void j_setshortcut (int obj , char chr);	
Argumente	obj	int
	chr	char
Beschreibung	Bestimmt das Zeichen chr als Shortcut für die Komponente .	
Zielkomponenten	MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu	

setsize

Synopsis	<code>void j_setsize (int obj , int width , int height);</code>						
Argumente	<table> <tr> <td><code>obj</code></td><td><code>int</code></td></tr> <tr> <td><code>width</code></td><td><code>int</code></td></tr> <tr> <td><code>height</code></td><td><code>int</code></td></tr> </table>	<code>obj</code>	<code>int</code>	<code>width</code>	<code>int</code>	<code>height</code>	<code>int</code>
<code>obj</code>	<code>int</code>						
<code>width</code>	<code>int</code>						
<code>height</code>	<code>int</code>						
Beschreibung	<p>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 beispielsweise 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.</p>						
Zielkomponenten	<p>Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment</p>						
Beispiel	<pre> : button = j_button(frame,"Button"); j_setsize(button,100,100); : </pre>						



setslidesize


Synopsis	int j_setslidesize (int obj , int val);	
Argumente	obj	int
	val	int
Beschreibung	Setzt die Größe des Schiebers auf den Wert von val .	
Zielkomponenten	Scrollbar	

setstate

Synopsis	void j_setstate (int obj , int bool);	
Argumente	obj	int
	bool	int
Beschreibung	Ist bool J_TRUE so wird die Komponente selectiert.	
Zielkomponenten	Checkbox, Radiobutton, Checkmenuitem, Led	

settext

Synopsis	<code>void j_settext (int obj , char* str);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>str</div> <div>char*</div> </div>
Beschreibung	<p>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.</p>
Zielkomponenten	<p>Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield</p>
Beispiel	<pre> : button = j_button(frame,"Hello World"); j_settext(button,"Goodbye"); : </pre>



setunitinc

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

setvalue

Synopsis	void j_setvalue (int obj , int val);	
Argumente	obj	int
	val	int
Beschreibung	Stellt den aktuellen Wert der Komponenten obj auf val .	
Zielkomponenten	Scrollbar, Progress, Meter, Sevensegment	

setvgap

Synopsis `void j_setvgap (int obj , int vgap);`

Argumente `obj` `int`
 `vgap` `int`

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

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



setxor

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

sevensegment

Synopsis `int j_sevensegment (int obj , int color);`

Argumente `obj` `int`
 `color` `int`

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);  
j_setvalue(seven,5);  
:
```



showpopup

Synopsis	void j_showpopup (int obj , int xpos , int ypos);	
Argumente	obj	int
	xpos	int
	ypos	int
Beschreibung	Zeigt die Komponente an der Koordinate (xpos,ypos) an.	
Zielkomponenten	Popupmenu	

show

Synopsis	<code>void j_show (int obj);</code>
Argumente	<code>obj</code> <code>int</code>
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	<code>int j_sleep (int msec);</code>
Argumente	<code>msec</code> <code>int</code>
Beschreibung	Unterbricht die Ausführung der Applikation für msec Millisekunden.

start

Synopsis

```
int 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 J_FALSE zurückgegeben

Beispiel

```
:  
if(j_start() != J_TRUE)  
{  
    printf("can't connect to JAPI Kernel\n");  
    exit(0);  
}  
:
```


sync

Synopsis

```
void j_sync ( );
```

Beschreibung

Synchronisiert die Applikation mit dem JAPI Kernel.

textarea

Synopsis `int j_textarea (int obj , int rows , int columns);`

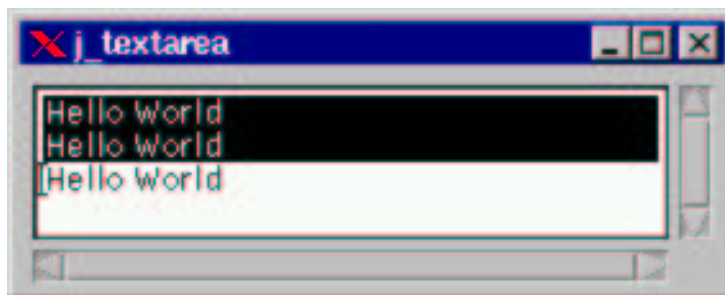
Argumente `obj` `int`
 `rows` `int`
 `columns` `int`

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	<code>int j_textfield (int obj , int columns);</code>
Argumente	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>columns</div> <div>int</div> </div>
Beschreibung	<p>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 <i>j_setsize()</i> verwendet werden.</p> <p>Ein Textfield liefert einen Event, sobald die Eingabe mit <i>Return</i> abgeschossen wird.</p>
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre> : frame = j_frame("j_textfield"); text = j_textfield(frame,30) : </pre>



translate

Synopsis `void j_translate (int obj , int x , int y);`

Argumente	obj	int
	x	int
	y	int

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

Zielkomponenten Canvas, Image, Printer

vscrollbar

Synopsis	int j_vscrollbar (int obj);
Argumente	obj int
Beschreibung	Diese Funktion erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück. Im Fehlerfall wird <code>-1</code> 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);
j_setpos(scroll,120,40);
j_setsize(scroll,20,100);
:

```



windowlistener

Synopsis	<code>int j_windowlistener (int window , int kind);</code>				
Argumente	<table><tr><td><code>window</code></td><td><code>int</code></td></tr><tr><td><code>kind</code></td><td><code>int</code></td></tr></table>	<code>window</code>	<code>int</code>	<code>kind</code>	<code>int</code>
<code>window</code>	<code>int</code>				
<code>kind</code>	<code>int</code>				
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 `int j_window (int obj);`

Argumente `obj` `int`

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 Fensterrahmen 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 ... ");
j_setnamedcolorbg(label,J_YELLOW);
:

```

Mouse pressed at 108:179