

japi

Reference Manual

Merten Joost

Inhaltsverzeichnis

I	Reference	5
1	Components	7
	Button	7
	Borderpanel	11
	Canvas	16
	Checkbox	22
	CheckmenuItem	25
	Choice	26
	Dialog	30
	Focuslistener	36
	Frame	37
	Helpmenu	44
	Hscrollbar	46
	Graphicbutton	50
	Graphiclabel	53
	Image	56
	KeyListener	59
	Label	60
	Led	63
	List	66
	Menu	70
	MenuItem	72
	Meter	73
	MouseListener	76
	Panel	77
	PopupMenu	82
	Printer	84
	Progressbar	87
	Radiobutton	90
	Sevensegment	93
	ScrollPane	96
	Textarea	99
	Textfield	103
	Vscrollbar	107
	Window	111

2 Functions	117
additem	117
add	119
alertbox	120
appendtext	121
beep	122
borderpanel	123
button	124
canvas	125
checkbox	126
checkmenuItem	127
choicebox2	128
choicebox3	129
choice	130
cliprect	131
componentlistener	132
connect	133
delete	134
deselect	135
dialog	136
disable	137
dispose	138
drawarc	139
drawcircle	140
drawimagesource	141
drawimage	142
drawline	143
drawoval	144
drawpixel	145
drawpolygon	146
drawpolyline	147
drawrect	148
drawroundrect	149
drawscaledimage	150
drawstring	151
enable	152
filedialog	153
fileselect	154
fillarc	155
fillcircle	156
filloval	157
fillpolygon	158
fillrect	159
fillroundrect	160
focuslistener	161
frame	162
getaction	163

getcolumns	164
getcurpos	165
getdanger	166
getfontascent	167
getfontheight	168
getheight	169
getimagesource	170
getimage	171
getinheight	172
getinsets	173
getinwidth	175
getitemcount	176
getitem	177
getkeychar	178
getkeycode	179
getlayoutid	180
getlength	181
getmousebutton	182
getmousex	183
getmousey	184
getparentid	185
getparent	186
getrows	187
getscaledimage	188
getscreenheight	189
getscreenwidth	190
getselect	191
getselend	192
getselstart	193
getseltext	194
getstate	195
getstringwidth	196
gettext	197
getvalue	198
getviewportheight	199
getviewportwidth	200
getwidth	201
getxpos	202
getypos	203
graphicbutton	204
graphiclabel	205
hasfocus	206
helpmenu	207
hide	208
hscrollbar	209
image	210
insert	211

inserttext	212
isparent	213
isresizable	214
isselect	215
isvisible	216
keylistener	217
label	218
led	219
line	220
list	222
loadimage	223
menubar	224
menuItem	225
menu	226
messagebox	227
meter	228
mouselistener	229
multiplemode	230
nextaction	231
pack	232
panel	233
popupmenu	234
printer	235
print	236
progressbar	237
quit	238
radiobutton	239
radiogroup	240
random	241
releaseall	242
release	243
removeall	244
removeitem	245
remove	246
replacetext	247
saveimage	248
scrollpane	249
selectall	250
select	251
selecttext	252
separator	253
setalign	254
setblockinc	255
setborderlayout	256
setborderpos	257
setcolorbg	258
setcolor	259

setcolumns	260
setcurpos	261
setcursor	262
setdanger	263
setdebug	264
setechochar	265
seteditable	266
setfixlayout	267
setflowfill	268
setflowlayout	269
setfocus	270
setfontname	271
setfontsize	272
setfontstyle	273
setFont	274
setgridlayout	276
sethgap	277
setIcon	278
setImage	279
setinsets	280
setMax	281
setmin	282
setnamedcolorbg	283
setnamedcolor	284
setnolayout	285
start	286
setpos	287
setradiogroup	288
setresizable	289
setrows	290
setshortcut	291
setsize	292
setslidesize	293
setstate	294
setText	295
setunitinc	296
setValue	297
setvgap	298
setxor	299
sevensegment	300
showpopup	301
show	302
sleep	303
start	304
sync	305
textarea	306
textfield	307

translate	308
vscrollbar	309
windowlistener	310
window	311

Teil I

Reference

Kapitel 1

Components

Button

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

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

Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.

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

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

Borderpanel

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

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

Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

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

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

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

Canvas

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

Zeichnet eine Linie in der momentan gesetzten Vordergrundfarbe von **(x1,y1)** nach **(x2,y2)**

- j_drawoval** *procedure j_drawoval (integer obj , integer x , integer y , integer rx , integer ry)*
 Zeichnet einen Oval mit dem Mittelpunkt **(x,y)** und den Hauptradien **rx** und **ry**.
- j_drawpixel** *procedure j_drawpixel (integer obj , integer x , integer y)*
 Zeichnet einen Pixel in der momentan gesetzten Vordergrundfarbe an die Position **(x,y)**.
- j_drawpolygon** *procedure j_drawpolygon (integer obj , integer len , sequence x , sequence y)*
 Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawpolyline** *procedure j_drawpolyline (integer obj , integer len , sequence x , sequence y)*
 Zeichnet eine Polylinie durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.
- j_drawrect** *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
 Zeichnet ein Rechteck an die Position **(x,y)** mit der Breite **width** und der Höhe **height**.
- j_drawroundrect** *procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
 Zeichnet ein Rechteck mit abgerundeten Ecken an die Position **(x,y)** mit der Breite **width** und der Höhe **height**. Die Radien der Ecken werden durch **arcx** und **arcy** bestimmt.
- j_drawscaledimage** *procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)*
 Kopiert einen Ausschnitt des Bildes **image**, der durch die Parameter **(sx, sy, sw, sh)** bestimmt wird an die Position **(tx, ty)** und scaliert den Ausschnitt auf die Breite **tw** und die Höhe **th**
- j_drawstring** *procedure j_drawstring (integer obj , integer x , integer y , sequence str)*
 Zeichnet den String **str** an die durch die Parameter **(x,y)** festgelegte Koordinate.
- j_enable** *procedure j_enable (integer obj)*
 enabled den Canvas **obj**.
- j_fillarc** *procedure j_fillarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*

Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt (\mathbf{x},\mathbf{y}) und den Hauptradien \mathbf{rx} und \mathbf{ry} . Die Parameter $\mathbf{arc1}$ und $\mathbf{arc2}$ bestimmen den Start- bzw. den Endwinkel des Kreisbogens.

- j_fillcircle** *procedure j_fillcircle* (*integer obj , integer x , integer y , integer r*)
Zeichnet einen gefülltes Kreis mit dem Mittelpunkt (\mathbf{x},\mathbf{y}) und dem Radius \mathbf{r} .
- j_filloval** *procedure j_filloval* (*integer obj , integer x , integer y , integer rx , integer ry*)
Zeichnet einen gefüllten Oval mit dem Mittelpunkt (\mathbf{x},\mathbf{y}) und den Hauptradien \mathbf{rx} und \mathbf{ry} .
- j_fillpolygon** *procedure j_fillpolygon* (*integer obj , integer len , sequence x , sequence y*)
Zeichnet ein gefülltes Polygon durch die Punkte, die durch die Arrays \mathbf{x} und \mathbf{y} gegeben sind.
- j_fillrect** *procedure j_fillrect* (*integer obj , integer x , integer y , integer width , integer height*)
Zeichnet ein gefülltes Rechteck an die Position (\mathbf{x},\mathbf{y}) mit der Breite \mathbf{width} und der Höhe \mathbf{height} .
- j_fillroundrect** *procedure j_fillroundrect* (*integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy*)
Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position (\mathbf{x},\mathbf{y}) . Das Rechteck hat eine Breite von \mathbf{width} und eine Höhe von \mathbf{height} Pixeln. Die Form der runden Ecken wird durch die Parameter \mathbf{arcx} und \mathbf{arcy} bestimmt.
- j_focuslistener** *function j_focuslistener* (*integer obj*)
Bindet an den Canvas \mathbf{obj} einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_getfontascent** *function j_getfontascent* (*integer obj*)
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Canvas \mathbf{obj} in Bildschirmpixeln zurück.
- j_getfontheight** *function j_getfontheight* (*integer obj*)
Liefert die totale Höhe des aktuellen Fonts des Canvas \mathbf{obj} in Bildschirmpixeln zurück.
- j_getheight** *function j_getheight* (*integer obj*)
Liefert die Höhe des Canvas \mathbf{obj} zurück.
- j_getimage** *function j_getimage* (*integer obj*)
Kopiert den Inhalt dem Canvas \mathbf{obj} in ein Image und liefert dessen Eventnummer zurück.
- j_getimagesource** *function j_getimagesource* (*integer obj , integer x , integer y , integer w , integer h , object r , object g , object b*)
Die Funktion liefert einen Ausschnitt vom Punkt (\mathbf{x}, \mathbf{y}) der Breite \mathbf{w} und der Höhe \mathbf{h} zurck. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays $\mathbf{r},\mathbf{g},\mathbf{b}$ abgelegt.

j_getparentid	<i>function</i> j_getparentid (<i>integer obj</i>) Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getparent	<i>function</i> j_getparent (<i>integer obj</i>) Liefert die Eventnummer des umgebenen Containers von obj zurück. Ist obj ein Frame wird -1 zurückgegeben.
j_getscaledimage	<i>function</i> j_getscaledimage (<i>integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th</i>) Kopiert den Inhalt des Ausschnitts von Positio (x , y) it der Breite sw und der höhe sh in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von tw und einer Höhe von th scaliert.
j_getstringwidth	<i>function</i> j_getstringwidth (<i>integer obj , sequence str</i>) Liefert die Länge des str des aktuellen Fonts des Canvas obj in Bildschirmpixeln zurück.
j_getwidth	<i>function</i> j_getwidth (<i>integer obj</i>) Liefert die Breite des Canvas obj zurück.
j_getxpos	<i>function</i> j_getxpos (<i>integer obj</i>) Liefert die aktuelle horizontale Position des Canvas obj zurück.
j_getypos	<i>function</i> j_getypos (<i>integer obj</i>) Liefert die aktuelle vertikale Position des Canvas obj zurück.
j_hide	<i>procedure</i> j_hide (<i>integer obj</i>) Zeigt den Canvas obj nicht mehr auf dem Bildschirm an.
j_isparent	<i>function</i> j_isparent (<i>integer obj , integer cont</i>) Liefert J_TRUE zurück, falls das cont die umgebende Komponente von obj ist, sonst J_FALSE
j_isvisible	<i>function</i> j_isvisible (<i>integer obj</i>) Liefert J_TRUE zurück, falls obj sichtbar ist, sonst J_FALSE
j_keylistener	<i>function</i> j_keylistener (<i>integer obj</i>) Bindet an den Canvas obj einen neuen Keylistener, und liefert dessen Eventnummer zurück.
j_mouselistener	<i>function</i> j_mouselistener (<i>integer obj , integer kind</i>) Bindet an den Canvas obj einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter kind bestimmt, wann ein Event ausgelöst wird.
j_popupmenu	<i>function</i> j_popupmenu (<i>integer obj , sequence label</i>) Erzeugt ein neues Popupmenu mit dem Inhalt label und liefert dessen Eventnummer zurück.
j_print	<i>procedure</i> j_print (<i>integer obj</i>) Druckt den Inhalt des Canvas .

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

Setzt den Ursprung des Koordinatensystems auf (\mathbf{x}, \mathbf{y}) .

Checkbox

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

Liefert J_TRUE , falls die Checkbox selektiert ist, sonst J_FALSE .

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

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

Checkmenüitem

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

Choice

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

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

Gibt die Choice **obj** aus dem umgebenden Container Element frei.

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

j_setsize *procedure j_setsize (integer obj , integer width , integer height)*
Setzt Breite und Höhe der Choice **obj** auf **width** and **height**.

j_show *procedure j_show (integer obj)*
Zeigt die Choice **obj** auf dem Bildschirm an.

Dialog

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

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

Erzeugt ein neues Graphiclabel mit dem Bildinhalt des Files, das durch **filename** spezifiziert wird.

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

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

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

j_show	<i>procedure</i> j_show (<i>integer obj</i>) Zeigt den Dialog obj auf dem Bildschirm an.
j_textarea	<i>function</i> j_textarea (<i>integer obj , integer rows , integer columns</i>) Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
j_textfield	<i>function</i> j_textfield (<i>integer obj , integer columns</i>) Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
j_vscrollbar	<i>function</i> j_vscrollbar (<i>integer obj</i>) Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
j_windowlistener	<i>function</i> j_windowlistener (<i>integer window , integer 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** *function* **j_focuslistener** (*integer obj*)
Bindet an den Focuslistener **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j_dispose** *procedure* **j_dispose** (*integer obj*)
der Focuslistener **obj** wird mit allen Recourcen gelöscht.
- j_hasfocus** *function* **j_hasfocus** (*integer obj*)
Liefert J.TRUE zurück, falls der Focuslistener den Focus besitzt, sonst J.FALSE .

Frame

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

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

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

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

Erzeugt eine LED der Farbe **color** und der Form **style**

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

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

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

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

j_window

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

Erzeugt ein neues Window und liefert dessen Eventnummer zurück.

Helpmenu

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

j_settext

procedure **j_settext** (*integer obj* , *sequence str*)
Ändert den Textinhalt des Helpmenu **obj** auf **str**

Hscrollbar

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

Liefert die Länge des **str** des aktuellen Fonts der Hscrollbar **obj** in Bildschirm-pixeln zurück.

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

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

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

Graphicbutton

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

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

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

Setzt den Cursor des Graphicbutton **obj** auf **cursor**.

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

Graphiclabel

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

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

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

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

Image

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

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

- j_fillrect** *procedure* **j_fillrect** (*integer obj , integer x , integer y , integer width , integer height*)
Zeichnet ein gefülltes Rechteck an die Position (**x,y**) mit der Breite **width** und der Höhe **height**.
- j_fillroundrect** *procedure* **j_fillroundrect** (*integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy*)
Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position (**x,y**). Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.
- j_getheight** *function* **j_getheight** (*integer obj*)
Liefert die Höhe des Image **obj** zurück.
- j_getimage** *function* **j_getimage** (*integer obj*)
Kopiert den Inhalt dem Image **obj** in ein Image und liefert dessen Eventnummer zurück.
- j_getimagesource** *function* **j_getimagesource** (*integer obj , integer x , integer y , integer w , integer h , object r , object g , object 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** *function* **j_getscaledimage** (*integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer 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** *function* **j_getwidth** (*integer obj*)
Liefert die Breite des Image **obj** zurück.
- j_print** *procedure* **j_print** (*integer obj*)
Druckt den Inhalt des Image .
- j_setxor** *procedure* **j_setxor** (*integer obj , integer bool*)
Schaltet den Painting Mode auf XOR, falls bool = J_TRUE ist. Im XOR Modus heben sich zwei gleiche Graphikbefehle auf.
- j_translate** *procedure* **j_translate** (*integer obj , integer x , integer y*)
Setzt den Ursprung des Koordinatensystems auf (**x, y**).

KeyListenerer

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

Label

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

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

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

Led

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

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

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

List

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

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

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

Setzt die Hintergrundfarbe auf eine vordefinierte Farbe **color**.

j_setnamedcolor *procedure* **j_setnamedcolor** (*integer obj , integer color*)
Setzt die Vordergrundfarbe auf eine vordefinierte Farbe **color**.

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

j_setsize *procedure* **j_setsize** (*integer obj , integer width , integer height*)
Setzt Breite und Höhe der List **obj** auf **width** and **height**.

j_show *procedure* **j_show** (*integer obj*)
Zeigt die List **obj** auf dem Bildschirm an.

Menu

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

Setzt den Fontgröße des Menu **obj** auf **size**.

j_setfontstyle *procedure* **j_setfontstyle** (*integer obj , integer style*)
Setzt den Fontstyle des Menu **obj** auf **style**.

j_setshortcut *procedure* **j_setshortcut** (*integer obj , integer chr*)
Bestimmt das Zeichen **chr** als Shortcut für das Menu .

j_settext *procedure* **j_settext** (*integer obj , sequence str*)
Ändert den Textinhalt des Menu **obj** auf **str**

Menuitem

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

Meter

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

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

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

Mouselistener

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

Panel

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

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

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

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

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

Setzt den Fontstyle des Panel **obj** auf **style**.

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

PopupMenu

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

j_settext *procedure* **j_settext** (*integer obj , sequence str*)
Ändert den Textinhalt des Popupmenu **obj** auf **str**

j_showpopup *procedure* **j_showpopup** (*integer obj , integer xpos , integer ypos*)
Zeigt das Popupmenu an der Koordinate (**xpos,ypos**) an.

Printer

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

Zeichnet ein Polygon durch die Punkte, die durch die Arrays **x** und **y** gegeben sind.

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

Zeichnet ein gefülltes Rechteck an die Position (\mathbf{x},\mathbf{y}) mit der Breite **width** und der Höhe **height**.

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

Zeichnet ein gefülltes Rechteck mit abgerundeten Ecken an die Position (\mathbf{x},\mathbf{y}) . Das Rechteck hat eine Breite von **width** und eine Höhe von **height** Pixeln. Die Form der runden Ecken wird durch die Parameter **arcx** und **arcy** bestimmt.

j_print *procedure* **j_print** (*integer obj*)

Druckt den Inhalt des Printer .

j_setxor *procedure* **j_setxor** (*integer obj , integer bool*)

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

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

Setzt den Ursprung des Koordinatensystems auf (\mathbf{x}, \mathbf{y}) .

Progressbar

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

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

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

Setzt den Cursor der Progressbar **obj** auf **cursor**.

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

Radiobutton

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

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

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

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

Sevenssegment

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

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

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

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

Scrollpane

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

Liefert die Länge des **str** des aktuellen Fonts der Scrollpane **obj** in Bildschirm-pixeln zurück.

- j_getviewportheight** *function* **j_getviewportheight** (*integer obj*)
Gibt die Höhe des sichtbaren Teils der Scrollpane **obj** zurück.
- j_getviewportwidth** *function* **j_getviewportwidth** (*integer obj*)
Gibt die Breite des sichtbaren Teils der Scrollpane **obj** zurück.
- j_getwidth** *function* **j_getwidth** (*integer obj*)
Liefert die Breite der Scrollpane **obj** zurück.
- j_getxpos** *function* **j_getxpos** (*integer obj*)
Liefert die aktuelle horizontale Position der Scrollpane **obj** zurück.
- j_getypos** *function* **j_getypos** (*integer obj*)
Liefert die aktuelle vertikale Position der Scrollpane **obj** zurück.
- j_hide** *procedure* **j_hide** (*integer obj*)
Zeigt die Scrollpane **obj** nicht mehr auf dem Bildschirm an.
- j_hscrollbar** *function* **j_hscrollbar** (*integer obj*)
Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.
- j_isparent** *function* **j_isparent** (*integer obj , integer cont*)
Liefert J_TRUE zurück, falls das **cont** die umgebende Komponente von **obj** ist, sonst J_FALSE
- j_isvisible** *function* **j_isvisible** (*integer obj*)
Liefert J_TRUE zurück, falls **obj** sichtbar ist, sonst J_FALSE
- j_keylistener** *function* **j_keylistener** (*integer obj*)
Bindet an die Scrollpane **obj** einen neuen Keylistener, und liefert dessen Eventnummer zurück.
- j_mouselistener** *function* **j_mouselistener** (*integer obj , integer kind*)
Bindet an die Scrollpane **obj** einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j_popupmenu** *function* **j_popupmenu** (*integer obj , sequence label*)
Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.
- j_print** *procedure* **j_print** (*integer obj*)
Druckt den Inhalt der Scrollpane .
- j_release** *procedure* **j_release** (*integer obj*)
Gibt die Scrollpane **obj** aus dem umgebenden Container Element frei.
- j_setborderpos** *procedure* **j_setborderpos** (*integer obj , integer pos*)

Positioniert die Scrollpane **obj** an die angegebene Position **pos**. Im umgebenden Container muß ein BorderLayout Manager eingestellt sein.

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

Textarea

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

Liefert die Höhe der Textarea **obj** zurück.

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

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

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

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

Textfield

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

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

Erzeugt ein neues Popupmenu mit dem Inhalt **label** und liefert dessen Eventnummer zurück.

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

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

Vscrollbar

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

Liefert die Länge des **str** des aktuellen Fonts der Vscrollbar **obj** in Bildschirm-pixeln zurück.

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

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

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

Window

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

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

Erzeugt eine horizontale Scrollbar und gibt dessen Eventnummer zurück.

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

Gibt alle Elemente aus dem Window **obj** frei.

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

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

Kapitel 2

Functions

additem

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

alertbox

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

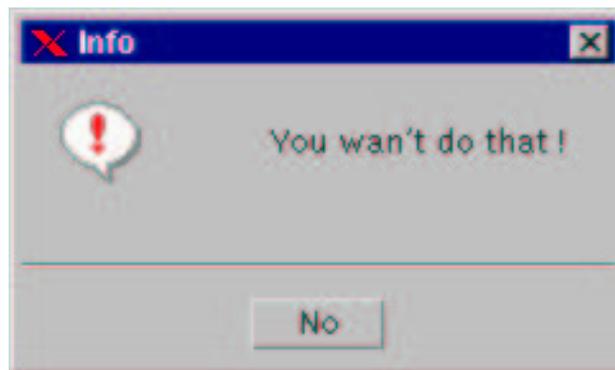
Argumente obj integer
 title sequence
 text sequence
 button sequence

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

Zielkomponenten Frame

Beispiel

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



appendtext

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

Argumente obj integer
 text sequence

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

Zielkomponenten Textarea

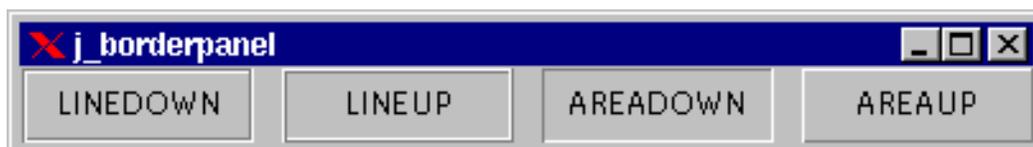
beep

Synopsis `procedure j_beep ()`

Beschreibung Es ertönt ein Systembeep.

borderpanel

Synopsis	function j_borderpanel (integer obj , integer type)
Argumente	obj integer type integer
Beschreibung	<p>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:</p> <ul style="list-style-type: none"> • 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. <p>Ein Borderpanel liefert bei einer Größenänderung ein Event.</p>
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	<pre> : 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) : </pre>



button

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

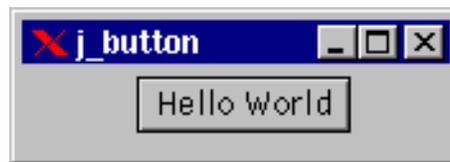
Argumente obj integer
 label sequence

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

Argumente obj integer
 width integer
 height integer

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

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



checkbox

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

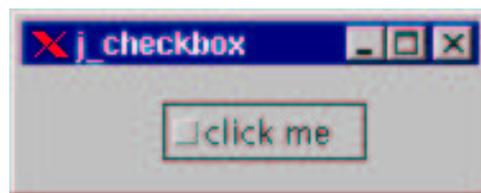
Argumente obj integer
 label sequence

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 function **j_checkmenuitem** (integer obj , sequence label)

Argumente obj integer
 label sequence

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

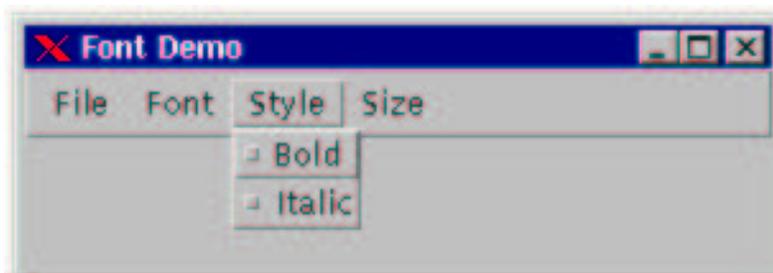
Zielkomponenten Menu, Popupmenu, Helpmenu

Beispiel

```

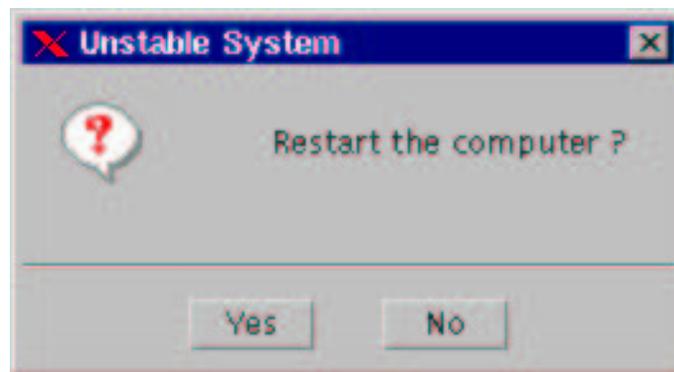
:
menubar = j_menubar(frame)
:
style = j_menu(menubar,"Style")
bold = j_checkmenuitem(style,"Bold")
italic= j_checkmenuitem(style,"Italic")
:

```



choicebox2

Synopsis	procedure j_choicebox2 (integer obj , sequence title , sequence text , sequence button1 , sequence button2)										
Argumente	<table border="0"> <tr><td>obj</td><td>integer</td></tr> <tr><td>title</td><td>sequence</td></tr> <tr><td>text</td><td>sequence</td></tr> <tr><td>button1</td><td>sequence</td></tr> <tr><td>button2</td><td>sequence</td></tr> </table>	obj	integer	title	sequence	text	sequence	button1	sequence	button2	sequence
obj	integer										
title	sequence										
text	sequence										
button1	sequence										
button2	sequence										
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	<pre> : retval = j_choicebox2(frame,"Unstable System","Restart the computer ?", " Yes ", "No") : </pre>										



checkbox3

Synopsis	procedure j_checkbox3 (integer obj , sequence title , sequence text , sequence button1 , sequence button2 , sequence button3)												
Argumente	<table border="0"> <tr><td>obj</td><td>integer</td></tr> <tr><td>title</td><td>sequence</td></tr> <tr><td>text</td><td>sequence</td></tr> <tr><td>button1</td><td>sequence</td></tr> <tr><td>button2</td><td>sequence</td></tr> <tr><td>button3</td><td>sequence</td></tr> </table>	obj	integer	title	sequence	text	sequence	button1	sequence	button2	sequence	button3	sequence
obj	integer												
title	sequence												
text	sequence												
button1	sequence												
button2	sequence												
button3	sequence												
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 function **j_choice** (integer obj)

Argumente obj integer

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

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

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



cliprect

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

componentlistener

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

connect

Synopsis function **j_connect** (sequence hostname)

Argumente hostname sequence

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

Beispiel

```
:
if( j_connect("atan.japi.de") = J_FALSE ) then

    or

if( j_connect("127.0.0.1") = J_FALSE ) then
:
```

delete

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

Argumente obj integer
 start integer
 end integer

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

Zielkomponenten Textarea

deselect

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

Argumente obj integer
 item integer

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

Zielkomponenten List

dialog

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

Argumente obj integer
 label sequence

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

Zielkomponenten Frame

Beispiel

```

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

```



disable

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

dispose

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

drawarc

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

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>rx</code>	<code>integer</code>
<code>ry</code>	<code>integer</code>
<code>arc1</code>	<code>integer</code>
<code>arc2</code>	<code>integer</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.

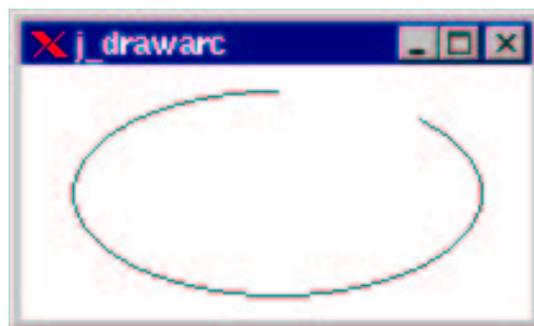
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



drawcircle

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

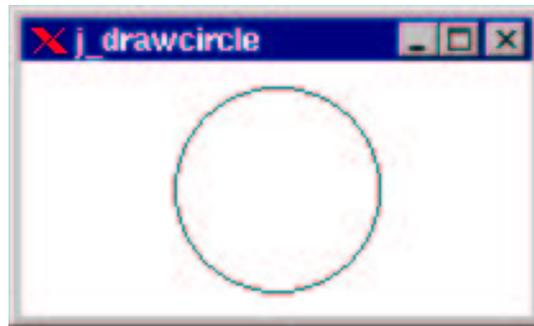
Argumente obj integer
 x integer
 y integer
 r integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

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



drawimagesource

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

drawimage

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

drawline

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

Argumente obj integer
 x1 integer
 y1 integer
 x2 integer
 y2 integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

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



drawoval

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

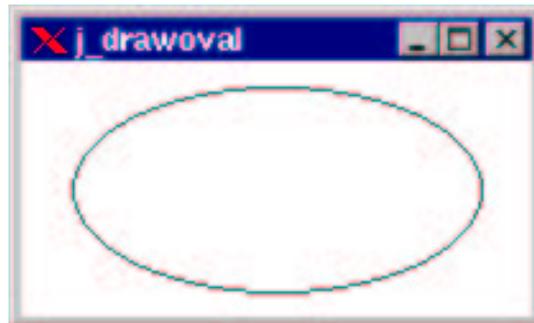
Argumente obj integer
 x integer
 y integer
 rx integer
 ry integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

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



drawpixel

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

Argumente obj integer
 x integer
 y integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

```

:
canvas = j_canvas(frame,256,50)
for i=0 to 1000
  j_drawpixel(canvas, mod(j_random(),256), mod(j_random(),50))
next i
:

```



drawpolygon

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

Argumente obj integer
 len integer
 x sequence
 y sequence

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.

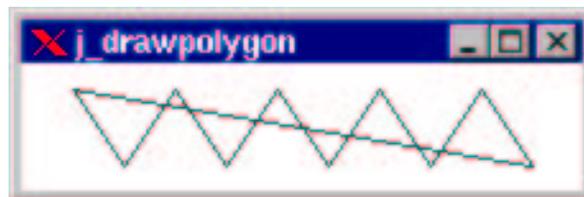
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



drawpolyline

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

Argumente obj integer
 len integer
 x sequence
 y sequence

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.

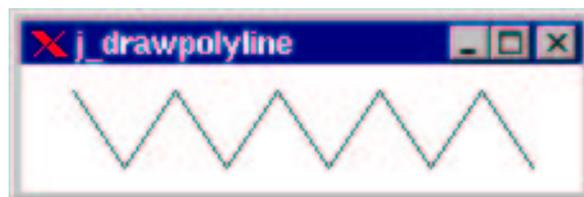
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



drawrect

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

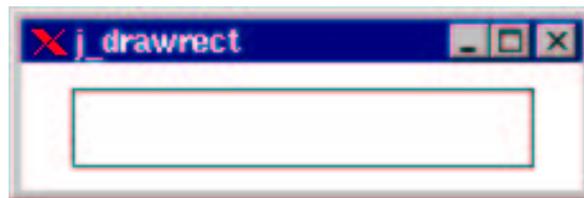
Argumente obj integer
 x integer
 y integer
 width integer
 height integer

Beschreibung Zeichnet 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 `procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)`

Argumente

<code>obj</code>	<code>integer</code>
<code>x</code>	<code>integer</code>
<code>y</code>	<code>integer</code>
<code>width</code>	<code>integer</code>
<code>height</code>	<code>integer</code>
<code>arcx</code>	<code>integer</code>
<code>arcy</code>	<code>integer</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.

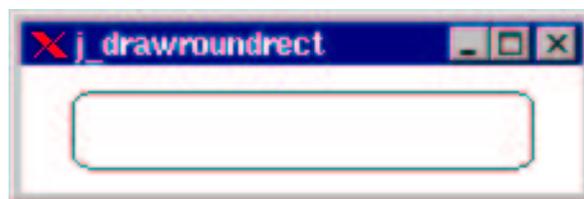
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



drawscaledimage

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

enable

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

filedialog

Synopsis function **j_filedialog** (integer frame , sequence title , sequence directory)

Argumente frame integer
 title sequence
 directory sequence

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.

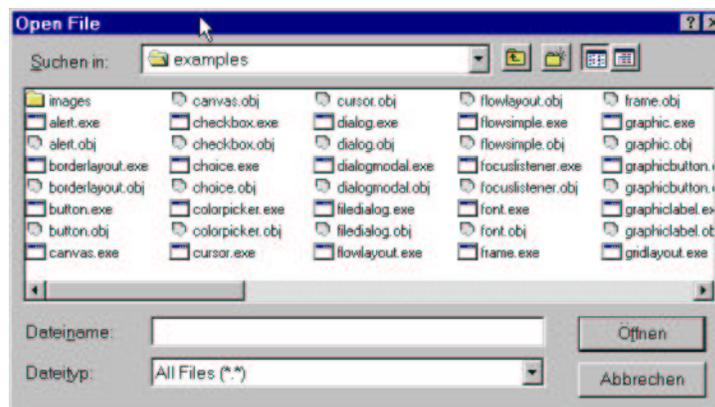
Zielkomponenten Frame

Beispiel

```

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

```



fileselect

Synopsis function **j_fileselect** (integer frame , sequence title , sequence filter)

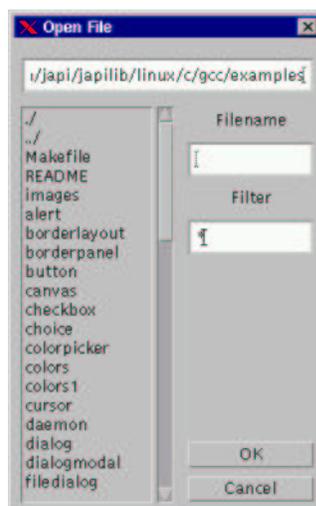
Argumente frame integer
 title sequence
 filter sequence

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

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



fillarc

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

Argumente

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

Beschreibung

Zeichnet einen gefüllten Kreisbogen mit dem Mittelpunkt **(x,y)** und den 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.

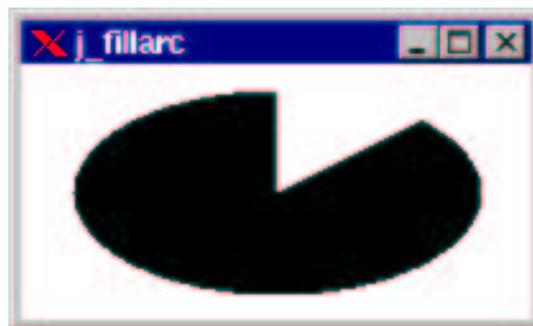
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



fillcircle

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

Argumente obj integer
 x integer
 y integer
 r integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

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



filloval

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

Argumente obj integer
 x integer
 y integer
 rx integer
 ry integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

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



fillpolygon

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

Argumente obj integer
len integer
x sequence
y sequence

Beschreibung Zeichnet eine gefülltes Polygon in der momentan gesetzten Vorder-
grundfarbe durch die Punkte, die durch die Arrays **x** und **y** ge-
geben sind. Beide Arrays müssen die mindestens **len** Elemente besit-
zen. Das Polygon wird automatisch geschlossen. Daher müssen die
erste und letzte Koordinate nicht übereinstimmen. Eingeschlosse-
ne Bereiche werden gefüllt dargestellt. Nullpunkt des Koordina-
tensystems ist die linke obere Ecke der Komponente die durch
obj referenziert wird.

Zielkomponenten Canvas, Image, Printer

Beispiel

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



fillrect

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

Argumente obj integer
 x integer
 y integer
 width integer
 height integer

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

Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



fillroundrect

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

Argumente

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

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.

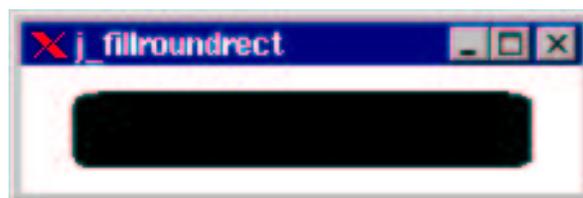
Zielkomponenten Canvas, Image, Printer

Beispiel

```

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

```



focuslistener

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

frame

Synopsis function **j_frame** (sequence label)

Argumente label sequence

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 function **j_getaction** ()

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

getcolumns

Synopsis procedure **j_getcolumns** (integer obj)

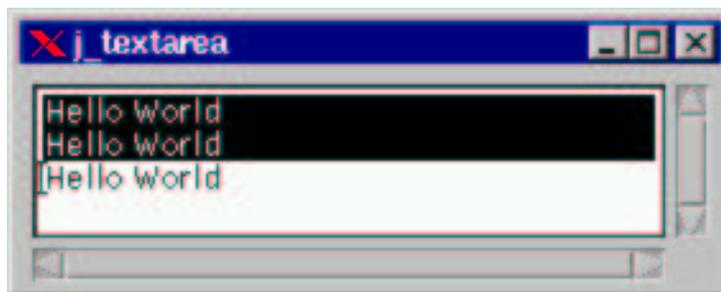
Argumente obj integer

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

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

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



getcurpos

Synopsis	function j_getcurpos (integer obj)
Argumente	obj integer
Beschreibung	Liefert die Position des Text Cursors zurück.
Zielkomponenten	Textarea, Textfield

getdanger

Synopsis procedure **j_getdanger** (integer obj)

Argumente obj integer

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

Zielkomponenten Meter

getfontascent

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

getfontheight

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

getheight

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

getimagesource

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

getimage

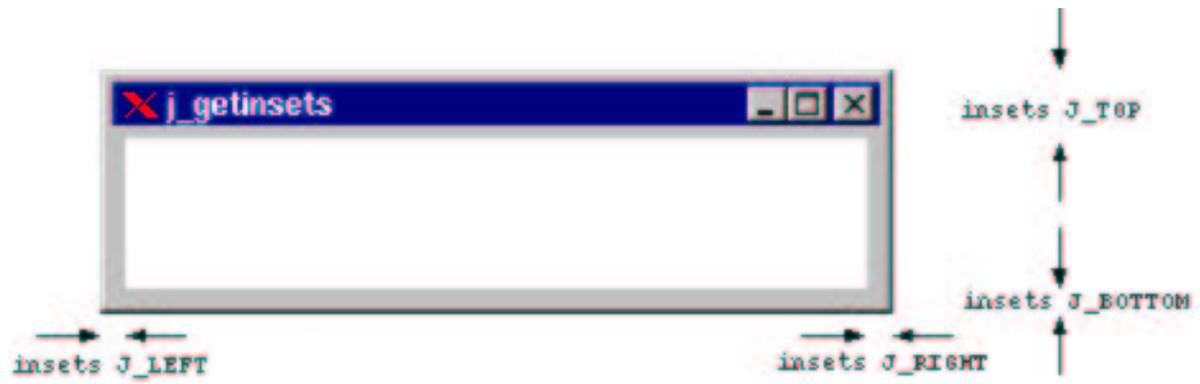
Synopsis	function j_getimage (integer obj)
Argumente	obj integer
Beschreibung	Kopiert den Inhalt der Komponenten obj in ein Image und liefert dessen Eventnummer zurück.
Zielkomponenten	Canvas, Image

getinheight

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

getinsets

Synopsis	function j_getinsets (integer obj , integer side)
Argumente	obj integer side integer
Beschreibung	<p>Liefert die angeforderte Breite des inneren Randes der Komponenten zurück. Der Parameter side kann folgende Werte annehmen:</p> <ul style="list-style-type: none"> • 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	<pre> : frame = j_frame("j_getinsets") print j_getinsets(frame,J_TOP) ,j_getinsets(frame,J_BOTTOM) , j_getinsets(frame,J_LEFT) ,j_getinsets(frame,J_RIGHT) : > 25 5 5 6 </pre>



getinwidth

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

getitemcount

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

getitem

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

getkeychar

Synopsis function **j_getkeychar** (integer obj)

Argumente obj integer

Beschreibung Liefert den ASCII Code, der zuletzt gedrückten Taste zurück.

Zielkomponenten Keylistener

getkeycode

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

getlayoutid

Synopsis function **j_getlayoutid** (integer obj)

Argumente obj integer

Beschreibung Liefert die Eventnummer des Layoutmanagers des Containers **obj** zurück.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

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

getlength

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

getmousebutton

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

getmousex

Synopsis	function j_getmousex (integer mouset listener)
Argumente	mouset listenerinteger
Beschreibung	Liefert die aktuelle horizontale Position der Mouse zurück.
Zielkomponenten	Mouset listener

getmousey

Synopsis	function j_getmousey (integer mouset listener)
Argumente	mouseteneinteger
Beschreibung	Liefert die aktuelle vertikale Position der Mouse zurück.
Zielkomponenten	Mouset listener

getparentid

Synopsis `function j_getparentid (integer obj)`

Argumente `obj` `integer`

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

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

Beispiel

```

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

```



getparent

Synopsis function **j_getparent** (integer obj)

Argumente obj integer

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

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

Beispiel

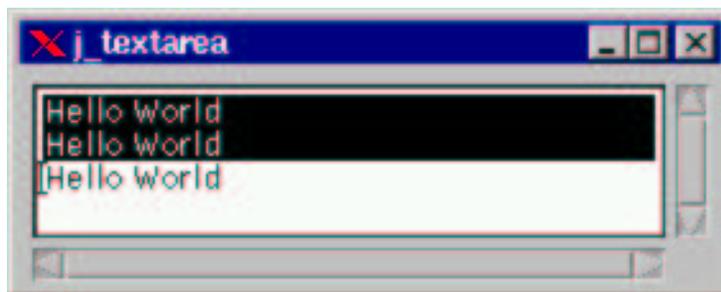
```
:
radio1       = j_radiobutton(j_radiogroup(frame),"Radiobutton 1")
radio2       = j_radiobutton(j_getparent(radio1),"Radiobutton 2")
:
```



getrows

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

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



getscaledimage

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

getscreenheight

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

getscreenwidth

Synopsis function **j_getscreenwidth** ()

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

Beispiel

```
:  
print j_getscreenwidth(), j_getscreenheight()  
:  
  
> 1280 1024
```

getselect

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

getselend

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

getselstart

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

getseltext

Synopsis function **j_getseltext** (integer obj)

Argumente obj integer

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

Zielkomponenten Textarea, Textfield

getstate

Synopsis	function j_getstate (integer obj)
Argumente	obj integer
Beschreibung	Liefert J_TRUE , falls die Komponente selektiert ist, sonst J_FALSE .
Zielkomponenten	Checkbox, Radiobutton, CheckmenuItem, Led

getstringwidth

Synopsis	function j_getstringwidth (integer obj , sequence str)
Argumente	obj integer str sequence
Beschreibung	Liefert die Länge des str 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

gettext

Synopsis	function j_gettext (integer obj)
Argumente	obj integer
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>: label = j_label(frame,"Hello World") print j_gettext(label) : > Hello World</pre>

getvalue

Synopsis function **j_getvalue** (integer obj)

Argumente obj integer

Beschreibung Liefert den aktuellen Wert der Komponenten .

Zielkomponenten Scrollbar

getviewportheight

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

getviewportwidth

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

getwidth

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

getxpos

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

getypos

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

graphiclabel

Synopsis	function j_graphiclabel (integer obj , sequence str)
Argumente	obj integer str sequence
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")
:

```

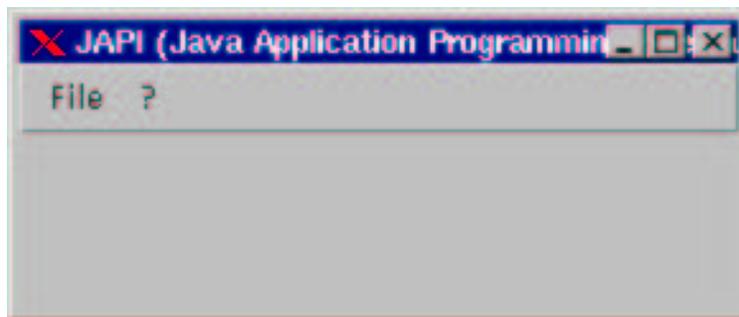


hasfocus

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

helpmenu

Synopsis	function j_helpmenu (integer obj , sequence label)
Argumente	obj integer label sequence
Beschreibung	Die Funktion erzeugt einen Hilfe-Menueintrag mit der Beschriftung label in der Komponente obj , und liefert dessen Eventnummer zurück. An diese Menueintrag können in weiteren Schritten MenuItem gebunden werden. Eine Menueintrag selbst erzeugt keinen Event. Hilfe Menü unterscheiden sich von normalen Menüs dadurch, daß sie je nach Oberfläche auch rechteckig erscheinen können. Zudem sind als Zielkomponenten nur Menubars zulässig.
Zielkomponenten	Menubar
Beispiel	<pre> : frame = j_frame("Menu Komponenten") menubar = j_menubar(frame) file= j_menu(menubar,"File") help= j_helpmenu(menubar,"?") : </pre>



hide

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

hscrollbar

Synopsis `function j_hscrollbar (integer obj)`

Argumente `obj integer`

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

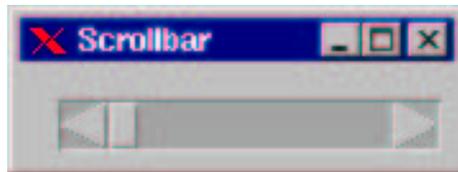
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame, Scrollpane

Beispiel

```

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

```



image

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

Argumente width integer
 height integer

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

Beispiel

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

insert

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

inserttext

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

isparent

Synopsis	function j_isparent (integer obj , integer cont)
Argumente	obj integer cont integer
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, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

isresizable

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

isselect

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

isvisible

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

label

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

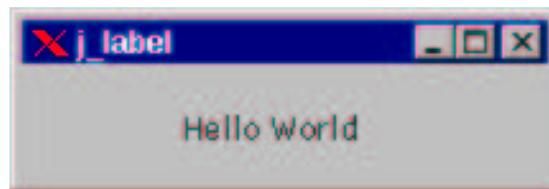
Argumente obj integer
 label sequence

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	function j_led (integer obj , integer style , integer color)						
Argumente	<table> <tr> <td>obj</td> <td>integer</td> </tr> <tr> <td>style</td> <td>integer</td> </tr> <tr> <td>color</td> <td>integer</td> </tr> </table>	obj	integer	style	integer	color	integer
obj	integer						
style	integer						
color	integer						
Beschreibung	Die Funktion erzeugt ein neues LED-Element und liefert die Eventnummer der Komponente zurück. Die LED kann eine runde Form besitzen style = J_ROUND oder eine eckige Form besitzen style=J_RECT . Die Farbe der LED kann eine der vordefinierten Farben sein (zB. J_RED, J_GREEN).						
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame						

Beispiel

```

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

```



line

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

Argumente obj integer
 orient integer
 style integer
 length integer

Beschreibung Die Funktion erzeugt ein Linienelement und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Das Element wird der Komponente zugeordnet, die durch **obj** referenziert wird. Die initiale Laenge der Linie wird durch den Parameter **length** bestimmt. Der Ruler dient der optischen Trennung anderer Komponenten.
 Der Parameter **orient** gibt an, ob eine senkrechte oder waagerechte Linie gezogen werden soll:

- J_HORIZONTAL : waagerechte Linie
- J_VERTICAL : senkrechte Linie

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

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

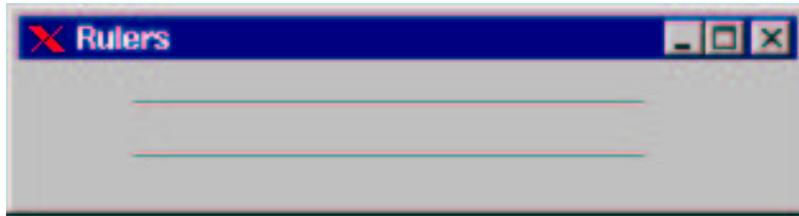
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



list

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

Argumente obj integer
 rows integer

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

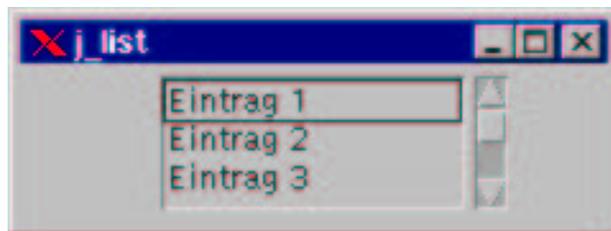
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



loadimage

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

menubar

Synopsis function **j_menubar** (integer obj)

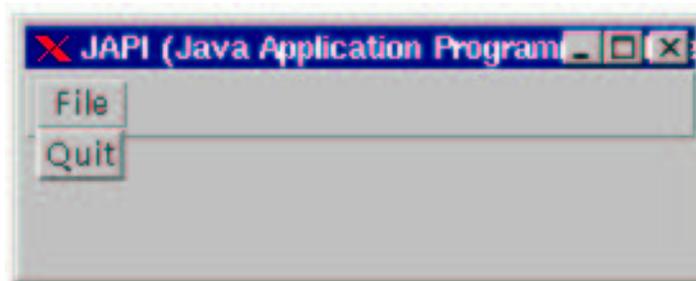
Argumente obj integer

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

Zielkomponenten Frame

Beispiel

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



menuitem

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

Argumente obj integer
 label sequence

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

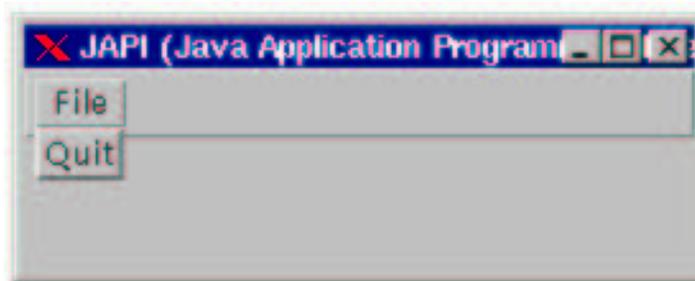
Zielkomponenten Menu, Popupmenu, Helpmenu

Beispiel

```

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

```



menu

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

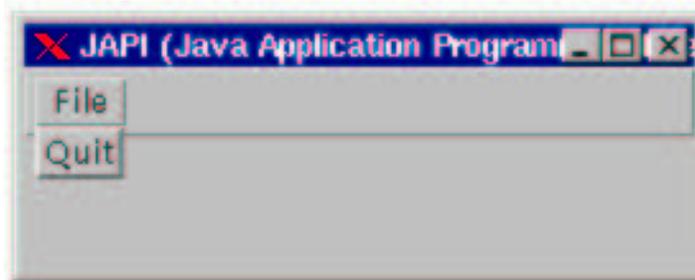
Argumente obj integer
 str sequence

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

Zielkomponenten Menubar, Menu

Beispiel

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



messagebox

Synopsis procedure **j_messagebox** (integer obj , sequence title , sequence text)

Argumente obj integer
 title sequence
 text sequence

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

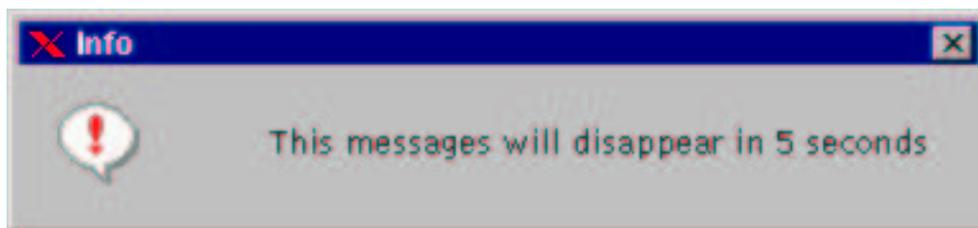
Zielkomponenten Frame

Beispiel

```

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

```



meter

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

Argumente obj integer
 title sequence

Beschreibung Die Funktion erzeugt ein analoges Anzeigeinstrument 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,40)
:

```



mouselistener

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

multiplemode

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

nextaction

Synopsis `function j_nextaction ()`

Beschreibung Wartet auf den nächsten Benutzerevent und liefert diesen zurück.

pack

Synopsis procedure **j_pack** (integer obj)

Argumente obj integer

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

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

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



panel

Synopsis function **j_panel** (integer obj)

Argumente obj integer

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

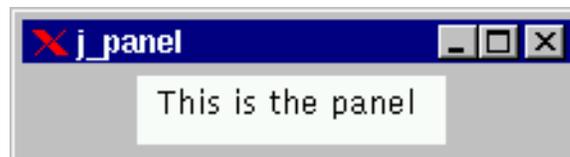
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



popupmenu

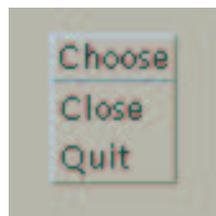
Synopsis	function j_popupmenu (integer obj , sequence label)
Argumente	obj integer label sequence
Beschreibung	Diese Funktion erzeugt ein neues Popupmenu und liefert dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Ein Popupmenu ist funktional identsch mit einem Menu, es ist allerdings an keine Menuleiste gebunden. Daher kann ein Popupmenu prinzipiell an jeder Position innerhalb der Oberfläche platziert werden. Wie an Menus kann auch an ein Popupmenu MenuItem's gebunden werden, die bei Anklicken des Popupmenus sichtbar werden.
Zielkomponenten	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```

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

```



printer

Synopsis	function j_printer (integer frame)
Argumente	frame integer
Beschreibung	Diese Funktion erzeugt ein neues Objekt, das eine Papierseite eines Druckes repräsentiert. Bis auf wenige Ausnahmen ist ein Printer Objekt funktional einem Canvas gleichgestellt. Die Funktion liefert die Eventnummer des Printers zurück, oder im Fehlerfall -1. Ein Printer erzeugt keinen Event.
Zielkomponenten	Frame
Beispiel	<pre>: printer = j_printer(frame) j_drawimage(printer,image,100,100) :</pre>

print

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



progressbar

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

Argumente obj integer
 orient integer

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

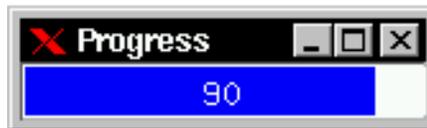
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



quit

Synopsis `procedure j_quit ()`

Beschreibung Beendet die Verbindung zum JAPI Kernel.

radiobutton

Synopsis	<code>function j_radiobutton (integer obj , sequence label)</code>				
Argumente	<table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td>integer</td> </tr> <tr> <td><code>label</code></td> <td>sequence</td> </tr> </table>	<code>obj</code>	integer	<code>label</code>	sequence
<code>obj</code>	integer				
<code>label</code>	sequence				
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 function **j_radiogroup** (integer obj)

Argumente obj integer

Beschreibung Diese Funktion erzeugt eine neue Radiogroup und liefert dessen Eventnummer zurück. Eine Radiogroup ist eine virtuelle Komponente, die dazu dient, Radiobutton funktionell zusammenzuschalten. Von allen Radiobutton, die in einer Radiogroup zusammengefasst sind, kann immer nur maximal ein Button aktiviert sein.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

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



random

Synopsis function **j_random** ()

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

releaseall

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

release

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

removeall

Synopsis function **j_removeall** (integer obj)

Argumente obj integer

Beschreibung entfernt alle Items aus der Komponenten .

Zielkomponenten List, Choice

removeitem

Synopsis function **j_removeitem** (integer obj , sequence item)

Argumente obj integer
 item sequence

Beschreibung entfernt den ersten Eintrag **item** aus der Komponenten .

Zielkomponenten List, Choice

remove

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

Argumente obj integer
 item integer

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

Zielkomponenten List, Choice

replacetext

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

saveimage

Synopsis	function j_saveimage (integer obj , sequence filename , integer filetype)
Argumente	obj integer filename sequence filetype integer
Beschreibung	Speichert den Inhalt der Komponenten in einem File mit dem Namen filename . Es werden zwei Fileformate unterstützt: <ul style="list-style-type: none">• BMP• PPM
Beispiel	<pre>: if(j_saveimage(canvas,"mandel.bmp",J_BMP) = J_FALSE) print "Error saving Bitmap file" :</pre>

scrollpane

Synopsis `function j_scrollpane (integer obj)`

Argumente `obj integer`

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

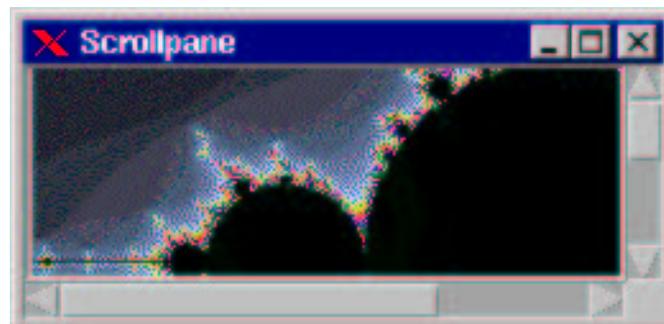
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



selectall

Synopsis procedure **j_selectall** (integer obj)

Argumente obj integer

Beschreibung Selectiert den gesamten Textinhalt der Komponenten .

Zielkomponenten Textarea, Textfield

select

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

Argumente obj integer
 item integer

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

Zielkomponenten List, Choice

selecttext

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

seperator

Synopsis `procedure j_seperator (integer obj)`

Argumente `obj integer`

Beschreibung Diese Prozedur erzeugt eine Trennlinie in einem Menu, das durch **obj** referenziert wird. Trennlinien dienen ausschließlich der optischen Trennung von MenuItemen, und besitzen keine weitere Funktionalität.

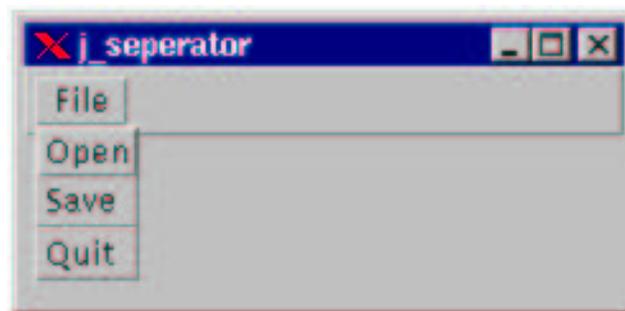
Zielkomponenten Menu, HelpMenu, Popupmenu

Beispiel

```

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

```



setalign

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

setblockinc

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

setborderlayout

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

setborderpos

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

setcolorbg

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

Argumente obj integer
 r integer
 g, integer
 b integer

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

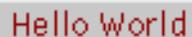
Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

Beispiel

```

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

```



setcolor

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

Argumente obj integer
 r integer
 g, integer
 b integer

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

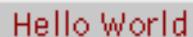
Zielkomponenten Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

Beispiel

```

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

```



Hello World

setcolumns

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

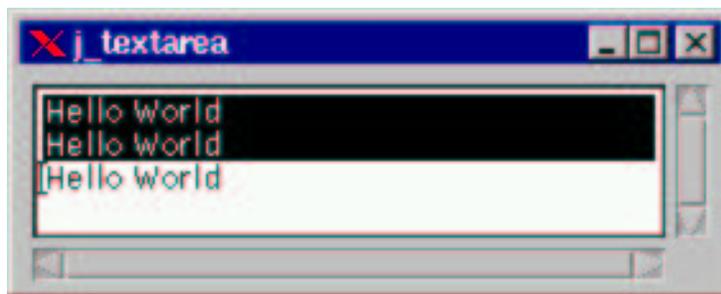
Argumente `obj` integer
 `columns` integer

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

Zielkomponenten Textarea, Textfield, Gridlayout

Beispiel

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



setcurpos

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

Argumente obj integer
 pos integer

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

Zielkomponenten Textarea, Textfield

setcursor

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

setdanger

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

Argumente obj integer
 val integer

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

Zielkomponenten Meter

setdebug

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

setechochar

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

seteditable

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

setfixlayout

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

setflowfill

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

setflowlayout

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

setfocus

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

setfontname

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

Beispiel

```

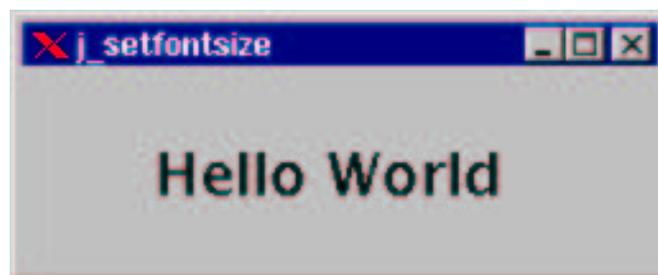
:
label = j_label(jframe,"Hello World")
j_setfontname(label,J_HELVETIA)
:

```



setfontsize

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



setfontstyle

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



setfont

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

Beispiel

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



setgridlayout

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

sethgap

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

Beispiel

```

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

```



seticon

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

Argumente frame integer
 icon integer

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

Zielkomponenten Frame

Beispiel

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

setimage

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

Argumente obj integer
 image integer

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

Zielkomponenten Graphicbutton, Graphiclabel

Beispiel

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



setinsets

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

Argumente obj integer
 top integer
 bottom integer
 left integer
 right integer

Beschreibung Setzt den inneren Rahmen auf die angegebenen Größen.

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

> 25 5 5 6

```



setmax

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

setmin

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

setnamedcolorbg

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

setnamedcolor

Synopsis procedure **j_setnamedcolor** (integer obj , integer color)

Argumente obj integer
 color integer

Beschreibung Setzt die Vordergrundfarbe auf eine vordefinierte Farbe **color**.

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

setnolayout

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

start

Synopsis procedure **j_start** (integer port)

Argumente port integer

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

Beispiel

```
:
j_setport(12345)
if(j_start() = J_FALSE) then
:
```

setpos

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

setradiogroup

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

setresizable

Synopsis procedure **j_setresizable** (integer obj , integer resizable)

Argumente obj integer
 resizable integer

Beschreibung Ist **resizable** J_FALSE , kann die Größe der Komponenten nicht vom Benutzer verändert werden.

Zielkomponenten Dialog, Frame

Beispiel

```
:  
frame = j_frame("fixsized Frame")  
j_setreziisable(frame,J_FALSE)  
:
```

setrows

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

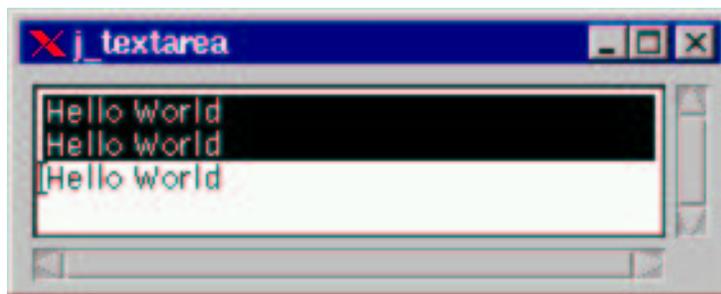
Argumente `obj` `integer`
 `rows` `integer`

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

Zielkomponenten Textarea, GridLayout

Beispiel

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



setshortcut

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

setsize

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

Argumente obj integer
 width integer
 height integer

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

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

Beispiel

```

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

```



setslidesize

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

Argumente obj integer
 val integer

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

Zielkomponenten Scrollbar

setstate

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

Argumente obj integer
 bool integer

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

Zielkomponenten Checkbox, Radiobutton, Checkmenueitem, Led

settext

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



setunitinc

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

Argumente obj integer
 val integer

Beschreibung Stellt die minimale Wertänderung von die Komponente auf den Wert von **val** ein.

Zielkomponenten Scrollbar

setvalue

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

Argumente obj integer
 val integer

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

Zielkomponenten Scrollbar, Progress, Meter, Sevensegment

setvgap

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

Argumente obj integer
 vgap integer

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

Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

Beispiel

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



setxor

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

sevenssegment

Synopsis	function j_sevenssegment (integer obj , integer color)
Argumente	obj integer color integer
Beschreibung	Die Funktion erzeugt eine Siebensegment-Anzeige und liefert die Eventnummer der Komponente zurück. Die Farbe der Siebensegment-Anzeige kann eine der vordefinierten Farben sein (zB. J_RED, J_GREEN).
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame
Beispiel	

```
:  
seven = j_sevenssegment(frame,J_GREEN)  
j_setvalue(seven,5)  
:
```



showpopup

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

show

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

sleep

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

start

Synopsis `function j_start ()`

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

Beispiel

```
:  
if(j_start() = J_FALSE) then  
  print "can't connect to JAPI Kernel"  
  exit  
end if  
:
```

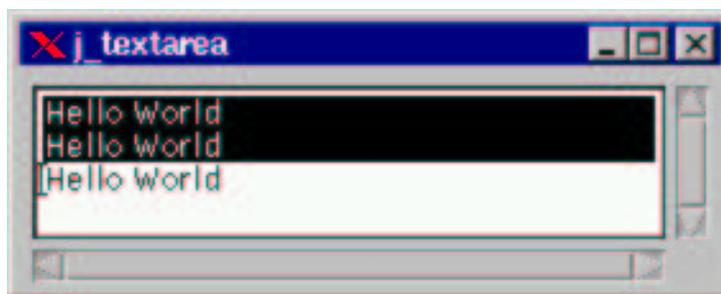
sync

Synopsis procedure **j_sync** ()

Beschreibung Synchronisiert die Applikation mit dem JAPI Kernel.

textarea

Synopsis	function j_textarea (integer obj , integer rows , integer columns)
Argumente	obj integer rows integer columns integer
Beschreibung	Diese Funktion erzeugt eine neue Textarea und gibt dessen Eventnummer zurück. Im Fehlerfall wird -1 zurückgegeben. Die Textarea hat eine sichtbare Größe von rows Reihen und columns Spalten, und ist somit abhängig vom voreingestellten Font. Soll eine Größe in Pixeleinheiten eingestellt werden, so kann nachträglich die Funktion <i>j_setsize()</i> 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	<pre> : frame = j_frame("j_textarea") text = j_textarea(frame,30,4) : </pre>



textfield

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

Beispiel

```

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

```



translate

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

Argumente obj integer
 x integer
 y integer

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

Zielkomponenten Canvas, Image, Printer

vscrollbar

Synopsis `function j_vscrollbar (integer obj)`

Argumente `obj integer`

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

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

Beispiel

```

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

```



windowlistener

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

window

Synopsis function **j_window** (integer obj)

Argumente obj integer

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

Zielkomponenten Frame

Beispiel

```

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

```

Mouse pressed at 108:179