

# japi

Reference Manual

Merten Joost



# Inhaltsverzeichnis

<b>I</b>	<b>Reference</b>	<b>5</b>
<b>1</b>	<b>Components</b>	<b>7</b>
	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

<b>2 Functions</b>	<b>117</b>
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 <b>obj</b> in Bildschirmpixeln zurück.
<b>j_getheight</b>	<i>function j_getheight ( integer obj )</i> Liefert die Höhe des Button <b>obj</b> zurück.
<b>j_getlength</b>	<i>function j_getlength ( integer obj )</i> Liefert die Länge des Textinhaltes den Button zurück.
<b>j_getparentid</b>	<i>function j_getparentid ( integer obj )</i> Liefert die Eventnummer des umgebenen Containers von <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getparent</b>	<i>function j_getparent ( integer obj )</i> Liefert die Eventnummer des umgebenen Containers von <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getstringwidth</b>	<i>function j_getstringwidth ( integer obj , sequence str )</i> Liefert die Länge des <b>str</b> des aktuellen Fonts des Button <b>obj</b> in Bildschirmpixeln zurück.
<b>j_gettext</b>	<i>function j_gettext ( integer obj )</i> Liefert den Textinhalt des Button .
<b>j_getwidth</b>	<i>function j_getwidth ( integer obj )</i> Liefert die Breite des Button <b>obj</b> zurück.
<b>j_getxpos</b>	<i>function j_getxpos ( integer obj )</i> Liefert die aktuelle horizontale Position des Button <b>obj</b> zurück.
<b>j_getypos</b>	<i>function j_getypos ( integer obj )</i> Liefert die aktuelle vertikale Position des Button <b>obj</b> zurück.
<b>j_hide</b>	<i>procedure j_hide ( integer obj )</i> Zeigt den Button <b>obj</b> nicht mehr auf dem Bildschirm an.
<b>j_isparent</b>	<i>function j_isparent ( integer obj , integer cont )</i> Liefert J_TRUE zurück, falls das <b>cont</b> die umgebende Komponente von <b>obj</b> ist, sonst J_FALSE
<b>j_isvisible</b>	<i>function j_isvisible ( integer obj )</i> Liefert J_TRUE zurück, falls <b>obj</b> sichtbar ist, sonst J_FALSE
<b>j_keylistener</b>	<i>function j_keylistener ( integer obj )</i> Bindet an den Button <b>obj</b> einen neuen Keylistener, und liefert dessen Eventnummer zurück.
<b>j_mouselistener</b>	<i>function j_mouselistener ( integer obj , integer kind )</i> Bindet an den Button <b>obj</b> einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> bestimmt, wann ein Event ausgelöst wird.
<b>j_popupmenu</b>	<i>function j_popupmenu ( integer obj , sequence label )</i>

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

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

**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.

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

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

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

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

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

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

	Setzt die Hintergrundfarbe auf den übergebenen RGB Wert
<b>j_setcolor</b>	<i>procedure</i> <b>j_setcolor</b> ( <i>integer obj , integer r , integer g , integer b</i> ) Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
<b>j_setcursor</b>	<i>function</i> <b>j_setcursor</b> ( <i>integer obj , integer cursor</i> ) Setzt den Cursor der Checkbox <b>obj</b> auf <b>cursor</b> .
<b>j_setfocus</b>	<i>function</i> <b>j_setfocus</b> ( <i>integer obj</i> ) Weist den Input Focus der Checkbox <b>obj</b> zu.
<b>j_setfontname</b>	<i>procedure</i> <b>j_setfontname</b> ( <i>integer obj , integer name</i> ) Setzt den Fonttyp der Checkbox <b>obj</b> auf <b>name</b> .
<b>j_setfont</b>	<i>procedure</i> <b>j_setfont</b> ( <i>integer obj , integer name , integer style , integer size</i> ) Setzt den Font der Checkbox <b>obj</b> entsprechend der Parameterliste.
<b>j_setfontsize</b>	<i>procedure</i> <b>j_setfontsize</b> ( <i>integer obj , integer size</i> ) Setzt den Fontgröße der Checkbox <b>obj</b> auf <b>size</b> .
<b>j_setfontstyle</b>	<i>procedure</i> <b>j_setfontstyle</b> ( <i>integer obj , integer style</i> ) Setzt den Fontstyle der Checkbox <b>obj</b> auf <b>style</b> .
<b>j_setnamedcolorbg</b>	<i>procedure</i> <b>j_setnamedcolorbg</b> ( <i>integer obj , integer color</i> ) Setzt die Hintergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setnamedcolor</b>	<i>procedure</i> <b>j_setnamedcolor</b> ( <i>integer obj , integer color</i> ) Setzt die Vordergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setpos</b>	<i>procedure</i> <b>j_setpos</b> ( <i>integer obj , integer xpos , integer ypos</i> ) Verschiebt die Checkbox <b>obj</b> zur angegebenen Koordinate ( <b>xpos,ypos</b> ) an.
<b>j_setsize</b>	<i>procedure</i> <b>j_setsize</b> ( <i>integer obj , integer width , integer height</i> ) Setzt Breite und Höhe der Checkbox <b>obj</b> auf <b>width</b> and <b>height</b> .
<b>j_setstate</b>	<i>procedure</i> <b>j_setstate</b> ( <i>integer obj , integer bool</i> ) Ist <b>bool</b> J_TRUE so wird die Checkbox selectiert.
<b>j_settext</b>	<i>procedure</i> <b>j_settext</b> ( <i>integer obj , sequence str</i> ) Ändert den Textinhalt der Checkbox <b>obj</b> auf <b>str</b>
<b>j_show</b>	<i>procedure</i> <b>j_show</b> ( <i>integer obj</i> ) Zeigt die Checkbox <b>obj</b> 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 <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getparent</b>	<i>function</i> <b>j_getparent</b> ( <i>integer obj</i> ) Liefert die Eventnummer des umgebenen Containers von <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getselect</b>	<i>function</i> <b>j_getselect</b> ( <i>integer obj</i> ) Liefert den Index des selectierten Items der Choice .
<b>j_getstringwidth</b>	<i>function</i> <b>j_getstringwidth</b> ( <i>integer obj , sequence str</i> ) Liefert die Länge des <b>str</b> des aktuellen Fonts der Choice <b>obj</b> in Bildschirmpixeln zurück.
<b>j_getwidth</b>	<i>function</i> <b>j_getwidth</b> ( <i>integer obj</i> ) Liefert die Breite der Choice <b>obj</b> zurück.
<b>j_getxpos</b>	<i>function</i> <b>j_getxpos</b> ( <i>integer obj</i> ) Liefert die aktuelle horizontale Position der Choice <b>obj</b> zurück.
<b>j_getypos</b>	<i>function</i> <b>j_getypos</b> ( <i>integer obj</i> ) Liefert die aktuelle vertikale Position der Choice <b>obj</b> zurück.
<b>j_hide</b>	<i>procedure</i> <b>j_hide</b> ( <i>integer obj</i> ) Zeigt die Choice <b>obj</b> nicht mehr auf dem Bildschirm an.
<b>j_insert</b>	<i>function</i> <b>j_insert</b> ( <i>integer obj , integer pos , sequence label</i> ) fügt ein neues Item in an der Position <b>pos</b> ein. Das Item trägt den Inhalt <b>label</b> .
<b>j_isparent</b>	<i>function</i> <b>j_isparent</b> ( <i>integer obj , integer cont</i> ) Liefert J_TRUE zurück, falls das <b>cont</b> die umgebende Komponente von <b>obj</b> ist, sonst J_FALSE
<b>j_isvisible</b>	<i>function</i> <b>j_isvisible</b> ( <i>integer obj</i> ) Liefert J_TRUE zurück, falls <b>obj</b> sichtbar ist, sonst J_FALSE
<b>j_keylistener</b>	<i>function</i> <b>j_keylistener</b> ( <i>integer obj</i> ) Bindet an die Choice <b>obj</b> einen neuen Keylistener, und liefert dessen Eventnummer zurück.
<b>j_mouselistener</b>	<i>function</i> <b>j_mouselistener</b> ( <i>integer obj , integer kind</i> ) Bindet an die Choice <b>obj</b> einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> bestimmt, wann ein Event ausgelöst wird.
<b>j_popupmenu</b>	<i>function</i> <b>j_popupmenu</b> ( <i>integer obj , sequence label</i> ) Erzeugt ein neues Popupmenu mit dem Inhalt <b>label</b> und liefert dessen Eventnummer zurück.
<b>j_print</b>	<i>procedure</i> <b>j_print</b> ( <i>integer obj</i> ) Druckt den Inhalt der Choice .
<b>j_release</b>	<i>procedure</i> <b>j_release</b> ( <i>integer obj</i> )

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

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

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

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

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

<b>j_show</b>	<i>procedure</i> <b>j_show</b> ( <i>integer obj</i> ) Zeigt den Dialog <b>obj</b> auf dem Bildschirm an.
<b>j_textarea</b>	<i>function</i> <b>j_textarea</b> ( <i>integer obj , integer rows , integer columns</i> ) Erzeugt eine neue Textarea und gibt dessen Eventnummer zurück.
<b>j_textfield</b>	<i>function</i> <b>j_textfield</b> ( <i>integer obj , integer columns</i> ) Erzeugt eine neues Textfield und gibt dessen Eventnummer zurück.
<b>j_vscrollbar</b>	<i>function</i> <b>j_vscrollbar</b> ( <i>integer obj</i> ) Erzeugt eine vertikale Scrollbar und gibt dessen Eventnummer zurück.
<b>j_windowlistener</b>	<i>function</i> <b>j_windowlistener</b> ( <i>integer window , integer kind</i> ) Bindet an den Dialog <b>window</b> einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> 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.

<b>j_disable</b>	<i>procedure</i> <b>j_disable</b> ( <i>integer obj</i> ) disabled den Frame <b>obj</b> , sodaß diese auf keine Benutzereingaben mehr reagiert.
<b>j_dispose</b>	<i>procedure</i> <b>j_dispose</b> ( <i>integer obj</i> ) der Frame <b>obj</b> wird mit allen Recourcen gelöscht.
<b>j_enable</b>	<i>procedure</i> <b>j_enable</b> ( <i>integer obj</i> ) enabled den Frame <b>obj</b> .
<b>j_filedialog</b>	<i>function</i> <b>j_filedialog</b> ( <i>integer frame , sequence title , sequence directory</i> ) Öffnet eine Filedialogbox in dem angegebenen <b>directory</b> mit dem Title <b>title</b> und liefert den ausgewählten Dateiname in <b>filename</b> zurück.
<b>j_fileselect</b>	<i>function</i> <b>j_fileselect</b> ( <i>integer frame , sequence title , sequence filter</i> ) Öffnet eine Fileslectorbox mit dem voreingestellten <b>filename</b> und dem Title <b>title</b> und liefert den ausgewählten Dateiname in <b>filename</b> zurück.
<b>j_focuslistener</b>	<i>function</i> <b>j_focuslistener</b> ( <i>integer obj</i> ) Bindet an den Frame <b>obj</b> einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
<b>j_getfontascent</b>	<i>function</i> <b>j_getfontascent</b> ( <i>integer obj</i> ) Liefert die Oberlänge (Ascent) des aktuellen Fonts des Frame <b>obj</b> in Bildschirmpixeln zurück.
<b>j_getfontheight</b>	<i>function</i> <b>j_getfontheight</b> ( <i>integer obj</i> ) Liefert die totale Höhe des aktuellen Fonts des Frame <b>obj</b> in Bildschirmpixeln zurück.
<b>j_getheight</b>	<i>function</i> <b>j_getheight</b> ( <i>integer obj</i> ) Liefert die Höhe des Frame <b>obj</b> zurück.
<b>j_getinheight</b>	<i>function</i> <b>j_getinheight</b> ( <i>integer cont</i> ) Liefert die Höhe des Frame <b>obj</b> des inneren Bereichs zurück.
<b>j_getinsets</b>	<i>function</i> <b>j_getinsets</b> ( <i>integer obj , integer side</i> ) Liefert die angeforderte Breite des inneren Randes dem Frame zurück.
<b>j_getinwidth</b>	<i>function</i> <b>j_getinwidth</b> ( <i>integer cont</i> ) Liefert die Breite des Frame <b>obj</b> des inneren Bereichs zurück.
<b>j_getlayoutid</b>	<i>function</i> <b>j_getlayoutid</b> ( <i>integer obj</i> ) Liefert die Eventnummer des Layoutmanagers des Containers <b>obj</b> zurück.
<b>j_getlength</b>	<i>function</i> <b>j_getlength</b> ( <i>integer obj</i> ) Liefert die Länge des Textinhaltes den Frame zurück.
<b>j_getparentid</b>	<i>function</i> <b>j_getparentid</b> ( <i>integer obj</i> ) Liefert die Eventnummer des umgebenen Containers von <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getparent</b>	<i>function</i> <b>j_getparent</b> ( <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 <b>color</b> und der Form <b>style</b>
<b>j_line</b>	<i>function</i> <b>j_line</b> ( <i>integer obj , integer orient , integer style , integer length</i> ) Erzeugt ein neues Linienelement und liefert dessen Eventnummer zurück
<b>j_list</b>	<i>function</i> <b>j_list</b> ( <i>integer obj , integer rows</i> ) Erzeugt eine neue List und gibt deren Eventnummer zurück.
<b>j_menubar</b>	<i>function</i> <b>j_menubar</b> ( <i>integer obj</i> ) Erzeugt eine neue Menubar in obj und liefert dessen Eventnummer zurück.
<b>j_messagebox</b>	<i>procedure</i> <b>j_messagebox</b> ( <i>integer obj , sequence title , sequence text</i> ) Öffnet eine Messagebox mit der Überschrift <b>title</b> und dem Textinhalt <b>text</b>
<b>j_meter</b>	<i>function</i> <b>j_meter</b> ( <i>integer obj , sequence title</i> ) Erzeugt ein analoges Anzeigeelement mit der Beschriftung <b>title</b> .
<b>j_mouselistener</b>	<i>function</i> <b>j_mouselistener</b> ( <i>integer obj , integer kind</i> ) Bindet an den Frame <b>obj</b> einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> bestimmt, wann ein Event ausgelöst wird.
<b>j_pack</b>	<i>procedure</i> <b>j_pack</b> ( <i>integer obj</i> ) Veranlaßt den Layoutmanager von <b>obj</b> die minimale Größe zu berechnen.
<b>j_panel</b>	<i>function</i> <b>j_panel</b> ( <i>integer obj</i> ) Erzeugt ein neues panel und liefert dessen Eventnummer zurück.
<b>j_popupmenu</b>	<i>function</i> <b>j_popupmenu</b> ( <i>integer obj , sequence label</i> ) Erzeugt ein neues Popuptaste mit dem Inhalt <b>label</b> und liefert dessen Eventnummer zurück.
<b>j_printer</b>	<i>function</i> <b>j_printer</b> ( <i>integer frame</i> ) Erzeugt ein neues Objekt, das eine Papierseite des Druckes repräsentiert.
<b>j_print</b>	<i>procedure</i> <b>j_print</b> ( <i>integer obj</i> ) Druckt den Inhalt des Frame .
<b>j_progressbar</b>	<i>function</i> <b>j_progressbar</b> ( <i>integer obj , integer orient</i> ) Erzeugt eine Progress Bar mit der Ausrichtung <b>orient</b> .
<b>j_radiogroup</b>	<i>function</i> <b>j_radiogroup</b> ( <i>integer obj</i> ) Erzeugt eine neue Radiogruppe und liefert dessen Eventnummer zurück.
<b>j_releaseall</b>	<i>procedure</i> <b>j_releaseall</b> ( <i>integer obj</i> ) Gibt alle Elemente aus dem Frame <b>obj</b> frei.
<b>j_release</b>	<i>procedure</i> <b>j_release</b> ( <i>integer obj</i> ) Gibt den Frame <b>obj</b> aus dem umgebenden Container Element frei.
<b>j_scrollpane</b>	<i>function</i> <b>j_scrollpane</b> ( <i>integer obj</i> ) Erzeugt eine neue Scrollpane und gibt dessen Eventnummer zurück.

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

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

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

**j\_window**

*function* **j\_window** ( *integer obj* )

Erzeugt ein neues Window und liefert dessen Eventnummer zurück.

Helpmenu
----------

<b>j_helpmenu</b>	<p><i>function</i> <b>j_helpmenu</b> ( <i>integer obj</i> , <i>sequence label</i> )          Erzeugt ein Hilfe-Menueintrag mit der Beschriftung <b>label</b> und liefert dessen Eventnummer zurück.</p>
<b>j_checkmenuitem</b>	<p><i>function</i> <b>j_checkmenuitem</b> ( <i>integer obj</i> , <i>sequence label</i> )          Erzeugt ein Checkmenuitem mit der Beschriftung <b>label</b> und liefert dessen Eventnummer zurück.</p>
<b>j_disable</b>	<p><i>procedure</i> <b>j_disable</b> ( <i>integer obj</i> )          disabled das Helpmenu <b>obj</b>, sodaß diese auf keine Benutzereingaben mehr reagiert.</p>
<b>j_dispose</b>	<p><i>procedure</i> <b>j_dispose</b> ( <i>integer obj</i> )          das Helpmenu <b>obj</b> wird mit allen Recourcen gelöscht.</p>
<b>j_enable</b>	<p><i>procedure</i> <b>j_enable</b> ( <i>integer obj</i> )          enabled das Helpmenu <b>obj</b>.</p>
<b>j_getlength</b>	<p><i>function</i> <b>j_getlength</b> ( <i>integer obj</i> )          Liefert die Länge des Textinhaltes das Helpmenu zurück.</p>
<b>j_gettext</b>	<p><i>function</i> <b>j_gettext</b> ( <i>integer obj</i> )          Liefert den Textinhalt des Helpmenu .</p>
<b>j_menuitem</b>	<p><i>function</i> <b>j_menuitem</b> ( <i>integer obj</i> , <i>sequence label</i> )          Erzeugt ein MenuItem mit der Beschriftung <b>label</b> und liefert dessen Eventnummer zurück.</p>
<b>j_seperator</b>	<p><i>procedure</i> <b>j_seperator</b> ( <i>integer obj</i> )          Erzeugt eine Trennlinie in das Helpmenu .</p>
<b>j_setfontname</b>	<p><i>procedure</i> <b>j_setfontname</b> ( <i>integer obj</i> , <i>integer name</i> )          Setzt den Fonttyp des Helpmenu <b>obj</b> auf <b>name</b>.</p>
<b>j_setfont</b>	<p><i>procedure</i> <b>j_setfont</b> ( <i>integer obj</i> , <i>integer name</i> , <i>integer style</i> , <i>integer size</i> )          Setzt den Font des Helpmenu <b>obj</b> entsprechend der Parameterliste.</p>
<b>j_setfontsize</b>	<p><i>procedure</i> <b>j_setfontsize</b> ( <i>integer obj</i> , <i>integer size</i> )          Setzt den Fontgröße des Helpmenu <b>obj</b> auf <b>size</b>.</p>
<b>j_setfontstyle</b>	<p><i>procedure</i> <b>j_setfontstyle</b> ( <i>integer obj</i> , <i>integer style</i> )          Setzt den Fontstyle des Helpmenu <b>obj</b> auf <b>style</b>.</p>
<b>j_setshortcut</b>	<p><i>procedure</i> <b>j_setshortcut</b> ( <i>integer obj</i> , <i>integer chr</i> )          Bestimmt das Zeichen <b>chr</b> 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
<b>j_setcolor</b>	<i>procedure</i> <b>j_setcolor</b> ( <i>integer obj , integer r , integer g , integer b</i> ) Setzt die Vordergrundfarbe auf den übergebenen RGB Wert
<b>j_setcursor</b>	<i>function</i> <b>j_setcursor</b> ( <i>integer obj , integer cursor</i> ) Setzt den Cursor der Hscrollbar <b>obj</b> auf <b>cursor</b> .
<b>j_setfocus</b>	<i>function</i> <b>j_setfocus</b> ( <i>integer obj</i> ) Weist den Input Focus der Hscrollbar <b>obj</b> zu.
<b>j_setfontname</b>	<i>procedure</i> <b>j_setfontname</b> ( <i>integer obj , integer name</i> ) Setzt den Fonttyp der Hscrollbar <b>obj</b> auf <b>name</b> .
<b>j_setfont</b>	<i>procedure</i> <b>j_setfont</b> ( <i>integer obj , integer name , integer style , integer size</i> ) Setzt den Font der Hscrollbar <b>obj</b> entsprechend der Parameterliste.
<b>j_setfontsize</b>	<i>procedure</i> <b>j_setfontsize</b> ( <i>integer obj , integer size</i> ) Setzt den Fontgröße der Hscrollbar <b>obj</b> auf <b>size</b> .
<b>j_setfontstyle</b>	<i>procedure</i> <b>j_setfontstyle</b> ( <i>integer obj , integer style</i> ) Setzt den Fontstyle der Hscrollbar <b>obj</b> auf <b>style</b> .
<b>j_setmax</b>	<i>function</i> <b>j_setmax</b> ( <i>integer obj , integer val</i> ) Stellt die obere Grenze der Hscrollbar auf den Wert von <b>val</b> ein.
<b>j_setmin</b>	<i>function</i> <b>j_setmin</b> ( <i>integer obj , integer val</i> ) Stellt die untere Grenze der Hscrollbar auf den Wert von <b>val</b> ein.
<b>j_setnamedcolorbg</b>	<i>procedure</i> <b>j_setnamedcolorbg</b> ( <i>integer obj , integer color</i> ) Setzt die Hintergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setnamedcolor</b>	<i>procedure</i> <b>j_setnamedcolor</b> ( <i>integer obj , integer color</i> ) Setzt die Vordergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setpos</b>	<i>procedure</i> <b>j_setpos</b> ( <i>integer obj , integer xpos , integer ypos</i> ) Verschiebt die Hscrollbar <b>obj</b> zur angegebenen Koordinate ( <b>xpos,ypos</b> ) an.
<b>j_setsize</b>	<i>procedure</i> <b>j_setsize</b> ( <i>integer obj , integer width , integer height</i> ) Setzt Breite und Höhe der Hscrollbar <b>obj</b> auf <b>width</b> and <b>height</b> .
<b>j_setslidesize</b>	<i>function</i> <b>j_setslidesize</b> ( <i>integer obj , integer val</i> ) Setzt die Größe des Schiebers auf den Wert von <b>val</b> .
<b>j_setunitinc</b>	<i>function</i> <b>j_setunitinc</b> ( <i>integer obj , integer val</i> ) Stellt die minimale Wertänderung von die Hscrollbar auf den Wert von <b>val</b> ein.
<b>j_setvalue</b>	<i>procedure</i> <b>j_setvalue</b> ( <i>integer obj , integer val</i> ) Stellt den aktuellen Wert der Hscrollbar <b>obj</b> auf <b>val</b> .
<b>j_show</b>	<i>procedure</i> <b>j_show</b> ( <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 <b>obj</b> auf <b>cursor</b> .
<b>j_setfocus</b>	<i>function</i> <b>j_setfocus</b> ( <i>integer obj</i> ) Weist den Input Focus dem Graphiclabel <b>obj</b> zu.
<b>j_setfontname</b>	<i>procedure</i> <b>j_setfontname</b> ( <i>integer obj , integer name</i> ) Setzt den Fonttyp des Graphiclabel <b>obj</b> auf <b>name</b> .
<b>j_setfont</b>	<i>procedure</i> <b>j_setfont</b> ( <i>integer obj , integer name , integer style , integer size</i> ) Setzt den Font des Graphiclabel <b>obj</b> entsprechend der Parameterliste.
<b>j_setfontsize</b>	<i>procedure</i> <b>j_setfontsize</b> ( <i>integer obj , integer size</i> ) Setzt den Fontgröße des Graphiclabel <b>obj</b> auf <b>size</b> .
<b>j_setfontstyle</b>	<i>procedure</i> <b>j_setfontstyle</b> ( <i>integer obj , integer style</i> ) Setzt den Fontstyle des Graphiclabel <b>obj</b> auf <b>style</b> .
<b>j_setimage</b>	<i>procedure</i> <b>j_setimage</b> ( <i>integer obj , integer image</i> ) Ordnet dem <b>obj</b> das Bild <b>image</b> zu.
<b>j_setnamedcolorbg</b>	<i>procedure</i> <b>j_setnamedcolorbg</b> ( <i>integer obj , integer color</i> ) Setzt die Hintergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setnamedcolor</b>	<i>procedure</i> <b>j_setnamedcolor</b> ( <i>integer obj , integer color</i> ) Setzt die Vordergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setpos</b>	<i>procedure</i> <b>j_setpos</b> ( <i>integer obj , integer xpos , integer ypos</i> ) Verschiebt das Graphiclabel <b>obj</b> zur angegebenen Koordinate ( <b>xpos,ypos</b> ) an.
<b>j_setsize</b>	<i>procedure</i> <b>j_setsize</b> ( <i>integer obj , integer width , integer height</i> ) Setzt Breite und Höhe des Graphiclabel <b>obj</b> auf <b>width</b> and <b>height</b> .
<b>j_show</b>	<i>procedure</i> <b>j_show</b> ( <i>integer obj</i> ) Zeigt das Graphiclabel <b>obj</b> 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**).

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



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

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

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

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

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

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

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

Meter
-------

- j\_meter**      *function* **j\_meter**    ( *integer obj , sequence title* )  
Erzeugt ein analoges Anzeigeeinstrument mit der Beschriftung **title**.
- j\_add**      *procedure* **j\_add**    ( *integer obj , integer cont* )  
Fügt das Anzeigeeinstrument **obj** in den Container **cont** ein.
- j\_componentlistener** *function* **j\_componentlistener**    ( *integer obj , integer kind* )  
Bindet an das Anzeigeeinstrument **obj** einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter **kind** bestimmt, wann ein Event ausgelöst wird.
- j\_disable**      *procedure* **j\_disable**    ( *integer obj* )  
disabled das Anzeigeeinstrument **obj**, sodaß diese auf keine Benutzereingaben mehr reagiert.
- j\_dispose**      *procedure* **j\_dispose**    ( *integer obj* )  
das Anzeigeeinstrument **obj** wird mit allen Recourcen gelöscht.
- j\_enable**      *procedure* **j\_enable**    ( *integer obj* )  
enabled das Anzeigeeinstrument **obj**.
- j\_focuslistener**      *function* **j\_focuslistener**    ( *integer obj* )  
Bindet an das Anzeigeeinstrument **obj** einen neuen Focuslistener, und liefert dessen Eventnummer zurück.
- j\_getdanger**      *procedure* **j\_getdanger**    ( *integer obj* )  
Liefert den Warnlevel des Anzeigeeinstrument **obj** zurück.
- j\_getfontascent**      *function* **j\_getfontascent**    ( *integer obj* )  
Liefert die Oberlänge (Ascent) des aktuellen Fonts des Anzeigeeinstrument **obj** in Bildschirmpixeln zurück.
- j\_getfontheight**      *function* **j\_getfontheight**    ( *integer obj* )  
Liefert die totale Höhe des aktuellen Fonts des Anzeigeeinstrument **obj** in Bildschirmpixeln zurück.
- j\_getheight**      *function* **j\_getheight**    ( *integer obj* )  
Liefert die Höhe des Anzeigeeinstrument **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.

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

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

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

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

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

**j\_filloval** *procedure* **j\_filloval** ( *integer obj , integer x , integer y , integer rx , integer ry* )

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

**j\_fillpolygon** *procedure* **j\_fillpolygon** ( *integer obj , integer len , 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 **(x,y)** mit der Breite **width** und der Höhe **height**.

**j\_fillroundrect**      *procedure* **j\_fillroundrect**    ( *integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy* )

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

**j\_print**              *procedure* **j\_print**    ( *integer obj* )

Druckt den Inhalt des Printer .

**j\_setxor**             *procedure* **j\_setxor**    ( *integer obj , integer bool* )

Schaltet den Painting Mode auf XOR, falls **bool = 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)**.

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 .

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

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

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

<b>j_setfocus</b>	<i>function</i> <b>j_setfocus</b> ( <i>integer obj</i> ) Weist den Input Focus der Textarea <b>obj</b> zu.
<b>j_setfontname</b>	<i>procedure</i> <b>j_setfontname</b> ( <i>integer obj</i> , <i>integer name</i> ) Setzt den Fonttyp der Textarea <b>obj</b> auf <b>name</b> .
<b>j_setfont</b>	<i>procedure</i> <b>j_setfont</b> ( <i>integer obj</i> , <i>integer name</i> , <i>integer style</i> , <i>integer size</i> ) Setzt den Font der Textarea <b>obj</b> entsprechend der Parameterliste.
<b>j_setfontsize</b>	<i>procedure</i> <b>j_setfontsize</b> ( <i>integer obj</i> , <i>integer size</i> ) Setzt den Fontgröße der Textarea <b>obj</b> auf <b>size</b> .
<b>j_setfontstyle</b>	<i>procedure</i> <b>j_setfontstyle</b> ( <i>integer obj</i> , <i>integer style</i> ) Setzt den Fontstyle der Textarea <b>obj</b> auf <b>style</b> .
<b>j_setnamedcolorbg</b>	<i>procedure</i> <b>j_setnamedcolorbg</b> ( <i>integer obj</i> , <i>integer color</i> ) Setzt die Hintergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setnamedcolor</b>	<i>procedure</i> <b>j_setnamedcolor</b> ( <i>integer obj</i> , <i>integer color</i> ) Setzt die Vordergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
<b>j_setpos</b>	<i>procedure</i> <b>j_setpos</b> ( <i>integer obj</i> , <i>integer xpos</i> , <i>integer ypos</i> ) Verschiebt die Textarea <b>obj</b> zur angegebenen Koordinate ( <b>xpos,ypos</b> ) an.
<b>j_setrows</b>	<i>procedure</i> <b>j_setrows</b> ( <i>integer obj</i> , <i>integer rows</i> ) Setzt die Anzahl Reihen von <b>obj</b> auf <b>rows</b> .
<b>j_setsize</b>	<i>procedure</i> <b>j_setsize</b> ( <i>integer obj</i> , <i>integer width</i> , <i>integer height</i> ) Setzt Breite und Höhe der Textarea <b>obj</b> auf <b>width</b> and <b>height</b> .
<b>j_settext</b>	<i>procedure</i> <b>j_settext</b> ( <i>integer obj</i> , <i>sequence str</i> ) Ändert den Textinhalt der Textarea <b>obj</b> auf <b>str</b>
<b>j_show</b>	<i>procedure</i> <b>j_show</b> ( <i>integer obj</i> ) Zeigt die Textarea <b>obj</b> 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 <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getparent</b>	<i>function j_getparent ( integer obj )</i> Liefert die Eventnummer des umgebenen Containers von <b>obj</b> zurück. Ist <b>obj</b> ein Frame wird $-1$ zurückgegeben.
<b>j_getselend</b>	<i>function j_getselend ( integer obj )</i> Liefert die Endposition des selectierten Textes.
<b>j_getselstart</b>	<i>function j_getselstart ( integer obj )</i> Liefert die Anfangsposition des selectierten Textes.
<b>j_getseltext</b>	<i>function j_getseltext ( integer obj )</i> Liefert den selectierten Text des Textfield <b>obj</b> zurück.
<b>j_getstringwidth</b>	<i>function j_getstringwidth ( integer obj , sequence str )</i> Liefert die Länge des <b>str</b> des aktuellen Fonts des Textfield <b>obj</b> in Bildschirm-pixeln zurück.
<b>j_gettext</b>	<i>function j_gettext ( integer obj )</i> Liefert den Textinhalt des Textfield .
<b>j_getwidth</b>	<i>function j_getwidth ( integer obj )</i> Liefert die Breite des Textfield <b>obj</b> zurück.
<b>j_getxpos</b>	<i>function j_getxpos ( integer obj )</i> Liefert die aktuelle horizontale Position des Textfield <b>obj</b> zurück.
<b>j_getypos</b>	<i>function j_getypos ( integer obj )</i> Liefert die aktuelle vertikale Position des Textfield <b>obj</b> zurück.
<b>j_hide</b>	<i>procedure j_hide ( integer obj )</i> Zeigt das Textfield <b>obj</b> nicht mehr auf dem Bildschirm an.
<b>j_isparent</b>	<i>function j_isparent ( integer obj , integer cont )</i> Liefert J_TRUE zurück, falls das <b>cont</b> die umgebende Komponente von <b>obj</b> ist, sonst J_FALSE
<b>j_isvisible</b>	<i>function j_isvisible ( integer obj )</i> Liefert J_TRUE zurück, falls <b>obj</b> sichtbar ist, sonst J_FALSE
<b>j_keylistener</b>	<i>function j_keylistener ( integer obj )</i> Bindet an das Textfield <b>obj</b> einen neuen Keylistener, und liefert dessen Eventnummer zurück.
<b>j_mouselistener</b>	<i>function j_mouselistener ( integer obj , integer kind )</i> Bindet an das Textfield <b>obj</b> einen neuen Mouselistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> bestimmt, wann ein Event ausgelöst wird.
<b>j_popupmenu</b>	<i>function j_popupmenu ( integer obj , sequence label )</i>

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

- j\_print**                    *procedure* **j\_print**    ( *integer obj* )  
Druckt den Inhalt des Textfield .
- j\_release**                *procedure* **j\_release**    ( *integer obj* )  
Gibt das Textfield **obj** aus dem umgebenden Container Element frei.
- j\_selectall**               *procedure* **j\_selectall**    ( *integer obj* )  
Selectiert den gesamten Textinhalt des Textfield .
- 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 das Textfield **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 des Textfield **obj** auf **cursor**.
- j\_setechochar**            *procedure* **j\_setechochar**    ( *integer obj , integer chr* )  
Bestimmt das Zeichen **chr** als Echo Zeichen für alle Benutzereingaben in das Textfield .
- j\_seteditable**             *procedure* **j\_seteditable**    ( *integer obj , integer bool* )  
Ist **bool** **J\_TRUE** , so ist das Textfield editierbar, ansonsten ist das Textfield nur lesbar für den Benutzer.
- j\_setfocus**              *function* **j\_setfocus**    ( *integer obj* )  
Weist den Input Focus dem Textfield **obj** zu.
- j\_setfontname**             *procedure* **j\_setfontname**    ( *integer obj , integer name* )  
Setzt den Fonttyp des Textfield **obj** auf **name**.
- j\_setfont**                 *procedure* **j\_setfont**    ( *integer obj , integer name , integer style , integer size* )  
Setzt den Font des Textfield **obj** entsprechend der Parameterliste.

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

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

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





## Kapitel 2

# Functions

additem
---------

Synopsis            procedure **j\_additem**    ( integer obj , sequence str )

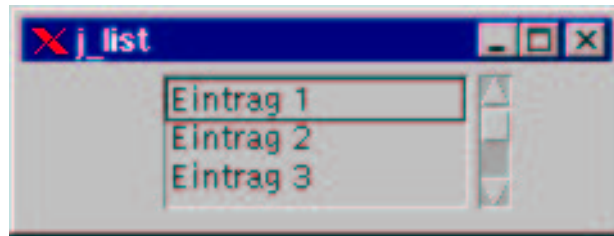
Argumente        obj            integer  
                  str            sequence

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 *j\_getselect()* ermittelt werden.

Zielkomponenten   List, Choice

Beispiel

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



add
-----

Synopsis	procedure <b>j_add</b> ( integer obj , integer cont )
Argumente	obj           integer cont           integer
Beschreibung	Fügt die Komponente <b>obj</b> in den Container <b>cont</b> 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 )

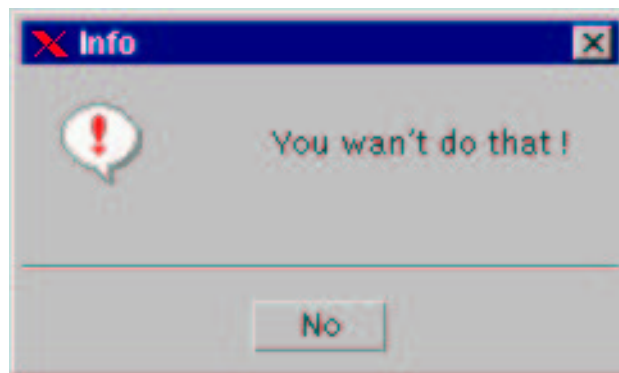
Argumente    obj            integer  
                  title          sequence  
                  text          sequence  
                  button        sequence

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

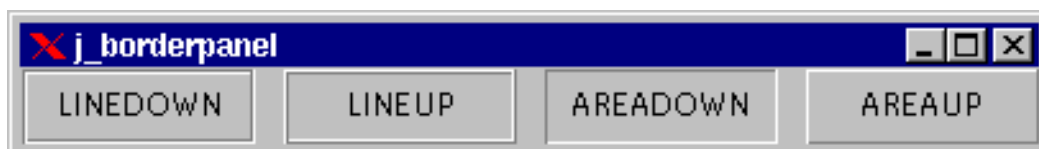
Synopsis	function <b>j_borderpanel</b> ( integer obj , integer type )
Argumente	obj            integer type           integer
Beschreibung	<p>Diese Funktion erzeugt ein neues Borderpanel und liefert dessen Eventnummer zurück. Im Fehlerfall wird <math>-1</math> zurückgegeben. Das Panel wird der Komponente zugeordnet, die durch <b>obj</b> 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"> <li>• J_LINEDOWN : Der Rahmen besteht aus einer Linie, die optisch abgesenkt erscheint.</li> <li>• J_LINEUP : Der Rahmen besteht aus einer Linie, die optisch hervorzutreten scheint.</li> <li>• J_AREADOWN : Das Panel scheint optisch abgesenkt.</li> <li>• J_AREAUP : Das Panel scheint optisch hervorzutreten.</li> </ul> <p>Ein Borderpanel liefert bei einer Größenänderung ein Event.</p>
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

### Beispiel

```

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

```



## button

Synopsis      function **j\_button** ( integer obj , sequence label )

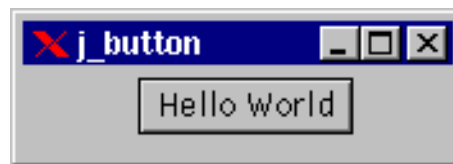
Argumente    obj            integer  
              label          sequence

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 )

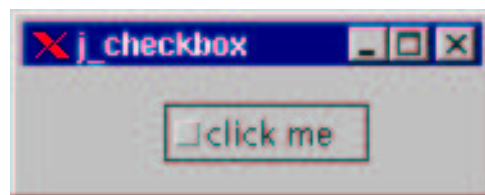
Argumente        obj            integer  
                  label          sequence

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.

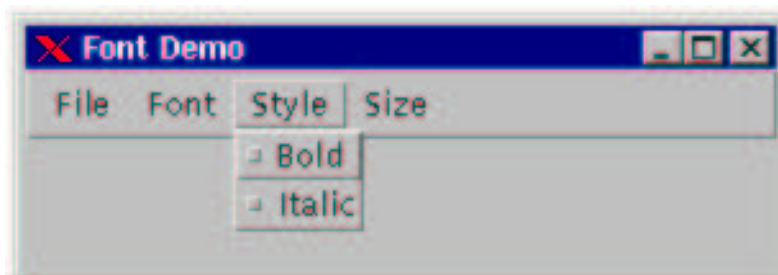
Zielkomponenten    Menu, Popupmenu, Helpmenu

Beispiel

```

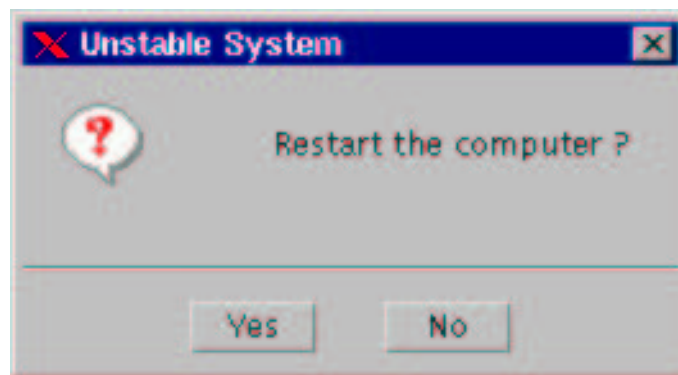
:
menubar = j_menubar(frame)
:
style = j_menu(menubar,"Style")
bold = j_checkmenuitem(style,"Bold")
italic= j_checkmenuitem(style,"Italic")
:

```



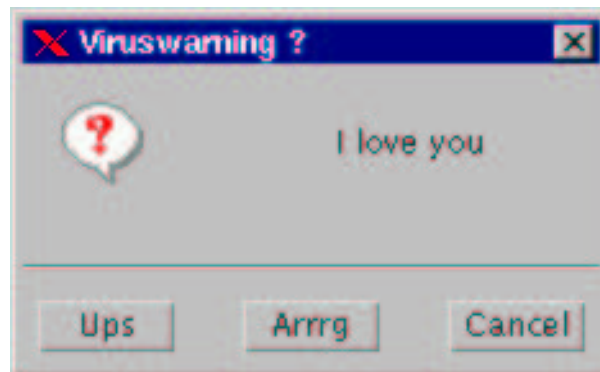
## checkbox2

Synopsis	procedure <b>j_checkbox2</b> ( 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 <b>title</b> , Textinhalt <b>text</b> 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_checkbox2(frame,"Unstable System","Restart the computer ?",                     " Yes ", "No") : </pre>										



## checkbox3

Synopsis	procedure <b>j_checkbox3</b> ( 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 <b>title</b> , Textinhalt <b>text</b> 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 )

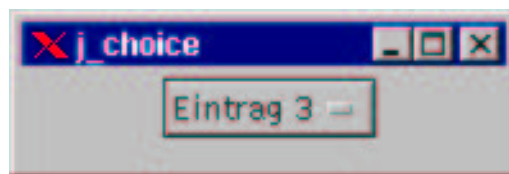
Argumente    obj            integer

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 <b>j_cliprect</b> ( 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 ( <b>x</b> , <b>y</b> , <b>width</b> , <b>height</b> ).
Zielkomponenten	Canvas, Image, Printer

componentlistener

Synopsis	function <b>j_componentlistener</b> ( integer obj , integer kind )
Argumente	obj            integer kind           integer
Beschreibung	Bindet an die Komponente <b>obj</b> einen neuen Componentlistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für <b>kind</b> sind: <ul style="list-style-type: none"><li>• J.RESIZED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> die Größe ändert.</li><li>• J.HIDDEN : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> verdeckt wird.</li><li>• J.SHOWN : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> sichtbar wird.</li></ul>
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 <b>j_connect</b> ( sequence hostname )
Argumente	hostname sequence
Beschreibung	Diese Funktion stellt eine Verbindung zu einem laufenden JAPI Kernel her. Der Kernel wird auf dem Rechner <b>hostname</b> gesucht. <b>hostname</b> 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 )

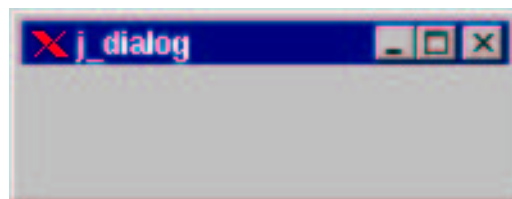
Argumente       obj           integer  
                 label        sequence

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

dispose
---------

Synopsis	procedure <b>j_dispose</b> ( integer obj )
Argumente	obj            integer
Beschreibung	die Komponente <b>obj</b> 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.

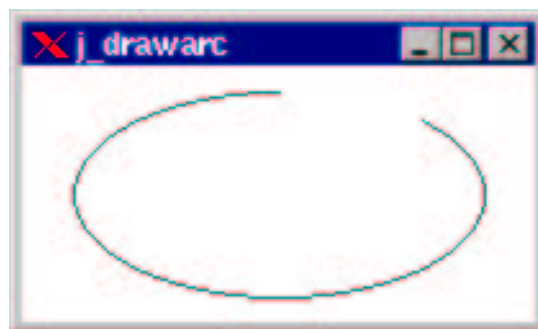
**Zielkomponenten**   Canvas, Image, Printer

**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 )

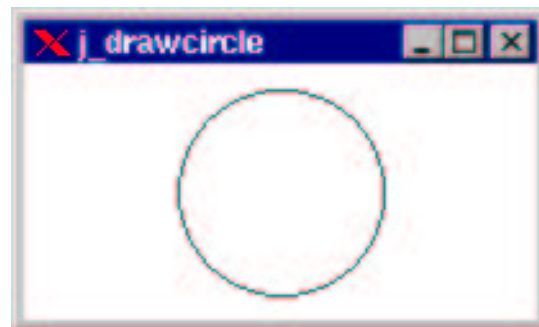
Argumente    obj            integer  
              x            integer  
              y            integer  
              r            integer

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 <b>j_drawimagesource</b> ( 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 ( <b>x,y</b> ) mit der Breite <b>w</b> und der Höhe <b>h</b> dar. Die Bilddaten werden als rot-, grün- und blau- Kanal in den Arrays r,g,b übergeben.
Zielkomponenten	Canvas, Image, Printer

drawimage
-----------

Synopsis	procedure <b>j_drawimage</b> ( integer obj , integer image , integer x , integer y )
Argumente	obj           integer image         integer x             integer y             integer
Beschreibung	Kopiert das Image mit der Eventnummer <b>image</b> an die Position ( <b>x</b> , <b>y</b> ).
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 )

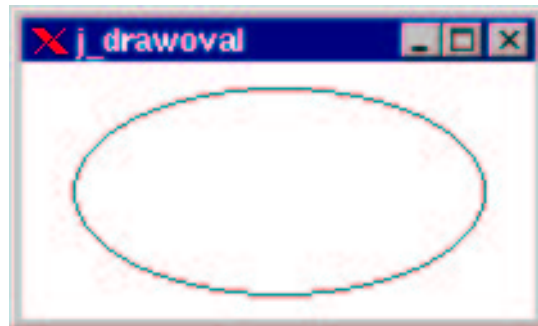
Argumente    obj            integer  
              x            integer  
              y            integer  
              rx           integer  
              ry           integer

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.

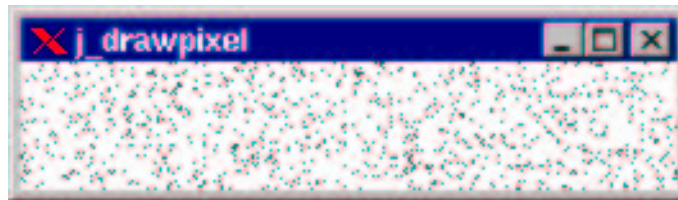
Zielkomponenten   Canvas, Image, Printer

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.

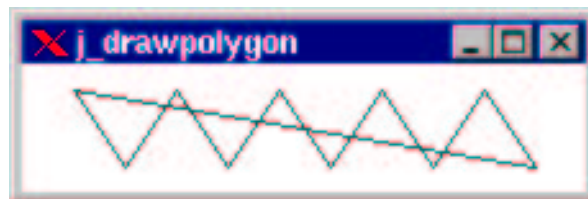
Zielkomponenten    Canvas, Image, Printer

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.

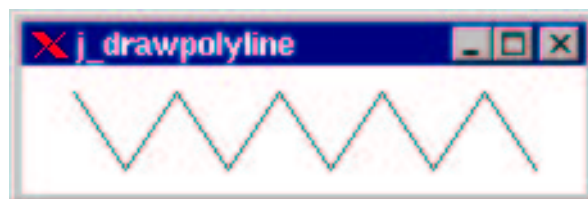
Zielkomponenten Canvas, Image, Printer

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 )

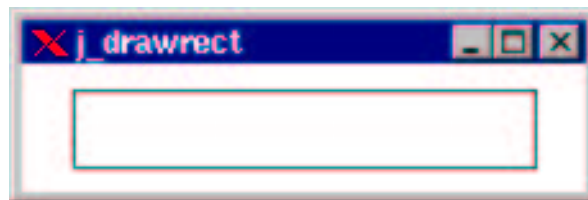
Argumente    obj            integer  
              x            integer  
              y            integer  
              width        integer  
              height       integer

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.

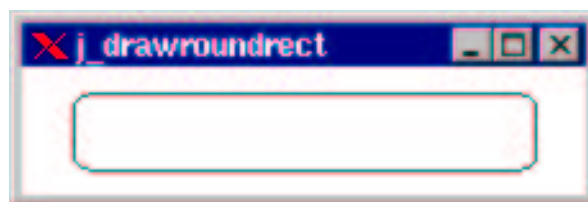
**Zielkomponenten**    Canvas, Image, Printer

**Beispiel**

```

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

```



drawscaledimage

Synopsis	procedure <b>j_drawscaledimage</b> ( 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 <b>image</b> , der durch die Parameter ( <b>sx</b> , <b>sy</b> , <b>sw</b> , <b>sh</b> ) bestimmt wird an die Position ( <b>tx</b> , <b>ty</b> ) und scaliert den Ausschnitt auf die Breite <b>tw</b> und die Höhe <b>th</b>
Zielkomponenten	Canvas, Image, Printer



enable
--------

Synopsis	procedure <b>j_enable</b> ( integer obj )
Argumente	obj            integer
Beschreibung	enabled die Komponente <b>obj</b> .
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.

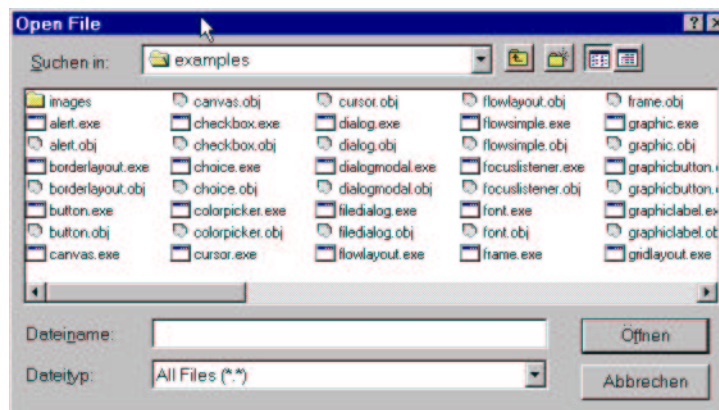
Zielkomponenten    Frame

Beispiel

```

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

```



fileselect

Synopsis           function **j\_fileselect** ( integer frame , sequence title , sequence filter )

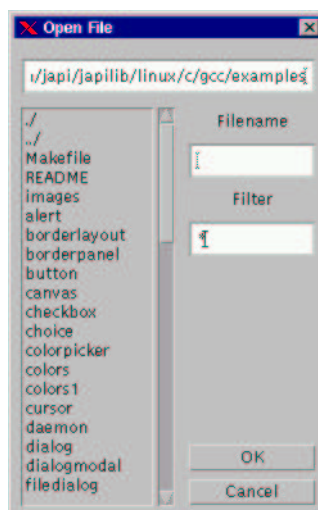
Argumente        frame        integer  
                   title        sequence  
                   filter       sequence

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.

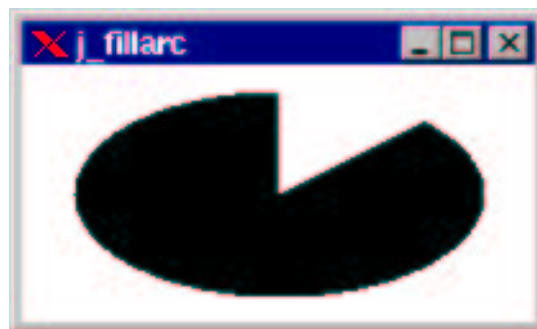
**Zielkomponenten**    Canvas, Image, Printer

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

**Zielkomponenten**    Canvas, Image, Printer

**Beispiel**

```

:
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.

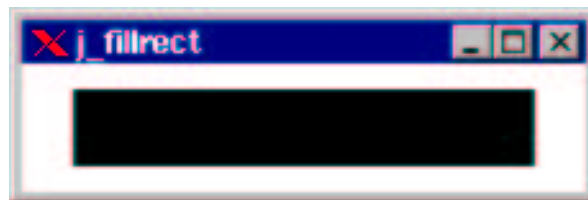
**Zielkomponenten**   Canvas, Image, Printer

**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.

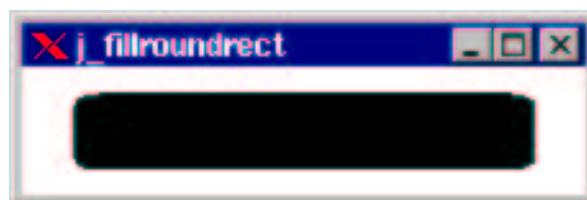
**Zielkomponenten**    Canvas, Image, Printer

**Beispiel**

```

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

```



focuslistener
---------------

Synopsis	function <b>j_focuslistener</b> ( integer obj )
Argumente	obj          integer
Beschreibung	Bindet an die Komponente <b>obj</b> 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

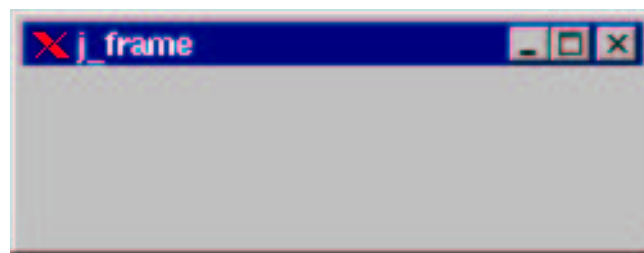
Synopsis      function **j\_frame** ( sequence label )

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 )`

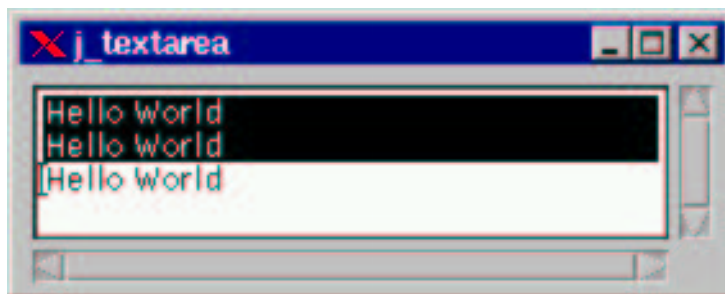
Argumente     `obj`        `integer`

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 <b>j_getcurpos</b> ( 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 <b>j_getfontascent</b> ( integer obj )
Argumente	obj          integer
Beschreibung	Liefert die Oberlänge (Ascent) des aktuellen Fonts der Komponenten <b>obj</b> 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 <b>j_getfontheight</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Liefert die totale Höhe des aktuellen Fonts der Komponenten <b>obj</b> 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 <b>j_getheight</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Die Funktion liefert die Höhe der Komponenten <b>obj</b> 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) : &gt; 22</pre>

getimagesource
----------------

Synopsis	function <b>j_getimagesource</b> ( 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 ( <b>x</b> , <b>y</b> ) der Breite <b>w</b> und der Höhe <b>h</b> zurück. Der Bildinhalt wird als rot-, grün- und blau- Kanal in den Arrays r,g,b abgelegt.
Zielkomponenten	Canvas, Image

getimage
----------

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

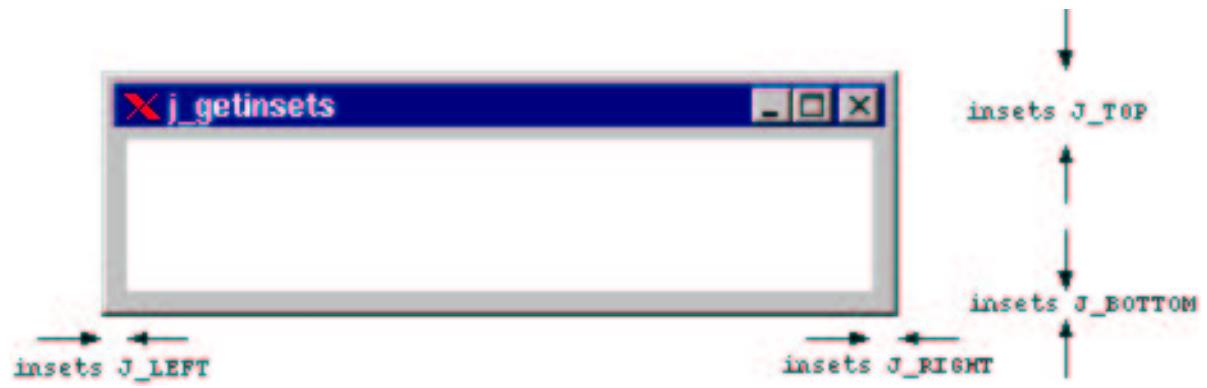
getinheight

Synopsis	function <b>j_getinheight</b> ( integer cont )
Argumente	cont          integer
Beschreibung	Die Funktion liefert die Höhe des inneren Bereichs der Komponenten <b>obj</b> 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)) : &gt; 370</pre>



getinsets
-----------

Synopsis	function <b>j_getinsets</b> ( integer obj , integer side )
Argumente	obj            integer side           integer
Beschreibung	<p>Liefert die angeforderte Breite des inneren Randes der Komponenten zurück. Der Parameter <b>side</b> kann folgende Werte annehmen:</p> <ul style="list-style-type: none"> <li>• J_TOP: die Höhe des oberen Rands wird zurueckgeliefert.</li> <li>• J_BOTTOM: die Höhe des unteren Rands wird zurueckgeliefert.</li> <li>• J_LEFT: die Breite des linken Rands wird zurueckgeliefert.</li> <li>• J_RIGHT: die Breite des rechten Rands wird zurueckgeliefert.</li> </ul>
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) : &gt; 25 5 5 6 </pre>



getinwidth
------------

Synopsis	function <b>j_getinwidth</b> ( integer cont )
Argumente	cont          integer
Beschreibung	Die Funktion liefert die Breite des inneren Bereich der Komponenten <b>obj</b> 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) : &gt; 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 <b>j_getitem</b> ( integer obj , integer item )
Argumente	obj            integer item           integer
Beschreibung	liefert den Inhalt des Items mit dem Index <b>item</b> 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 <b>j_getkeycode</b> ( 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 <b>j_getlength</b> ( 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 <b>j_getmousebutton</b> ( integer mousetlistener )
Argumente	mousetlisteneinteger
Beschreibung	Liefert die letzte gedrückte Mousetaste zurück. Der Rückgabewert ist <ul style="list-style-type: none"><li>• J_LEFT linke Mousetaste</li><li>• J_CENTER mittlere Mousetaste</li><li>• J_RIGHT rechte Mousetaste</li></ul>
Zielkomponenten	Mousetlistener

getmousex
-----------

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

getmousey
-----------

Synopsis	function <b>j_getmousey</b> ( 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

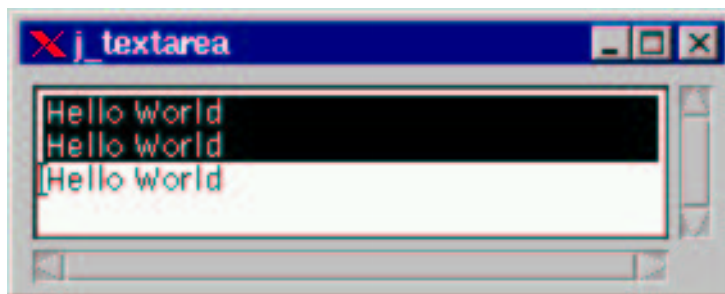
```
:
radio1       = j_radiobutton(j_radiogroup(frame),"Radiobutton 1")
radio2       = j_radiobutton(j_getparent(radio1),"Radiobutton 2")
:
```



## getrows

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

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



getscaledimage

Synopsis	function <b>j_getscaledimage</b> ( 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 ( <b>x</b> , <b>y</b> ) it der Breite <b>sw</b> und der höhe <b>sh</b> in ein Image und liefert dessen Eventnummer zurück. Das Image wird dabei auf eine Breite von <b>tw</b> und einer Höhe von <b>th</b> 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

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

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 <b>j_getselend</b> ( integer obj )
Argumente	obj          integer
Beschreibung	Liefert die Endposition des selectierten Textes.
Zielkomponenten	Textarea, Textfield

getselstart
-------------

Synopsis	function <b>j_getselstart</b> ( 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 <b>j_getstate</b> ( 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 <b>j_getstringwidth</b> ( integer obj , sequence str )
Argumente	obj            integer str            sequence
Beschreibung	Liefert die Länge des <b>str</b> des aktuellen Fonts der Komponenten <b>obj</b> 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 <b>j_gettext</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Liefert den Textinhalt der, durch <b>obj</b> referenzierten, Komponenten im Parameter <b>str</b> zurück. In C und in Pascal wird ein Pointer auf <b>str</b> 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) :  &gt; 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 <b>j_getviewportheight</b> ( integer obj )
Argumente	obj          integer
Beschreibung	Gibt die Höhe des sichtbaren Teils der Komponenten <b>obj</b> zurück.
Zielkomponenten	Scrollpane

getviewportwidth

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

## getwidth

Synopsis	function <b>j_getwidth</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Die Funktion liefert die Breite der Komponenten <b>obj</b> 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) : &gt; 84</pre>

getxpos
---------

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

## graphicbutton

Synopsis      `function j_graphicbutton` ( integer obj , sequence filename )

Argumente    obj            integer  
               filename      sequence

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

Zielkomponenten    Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

:  

frame = j_frame("j_graphicbutton")  

button = j_graphicbutton(frame,"save.gif")  

:

```





## graphiclabel

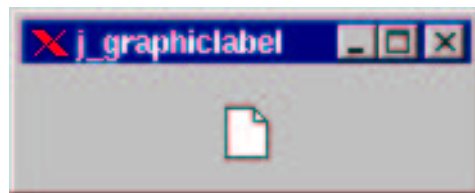
Synopsis	<code>function <b>j_graphiclabel</b> ( integer obj , sequence str )</code>				
Argumente	<table> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td>integer</td> </tr> <tr> <td><code>str</code></td> <td>sequence</td> </tr> </table>	<code>obj</code>	integer	<code>str</code>	sequence
<code>obj</code>	integer				
<code>str</code>	sequence				
Beschreibung	Die Funktion erzeugt ein neues Label mit einem Icon als Inhalt. Die Datei, die das Icon enthält, ist im Parameter <b>filename</b> anzugeben. Das Icon muß im GIF oder JPEG-Format vorliegen. Die Funktion liefert die Eventnummer des Labels zurück, oder im Fehlerfall <code>-1</code> .				
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame				

### Beispiel

```

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

```

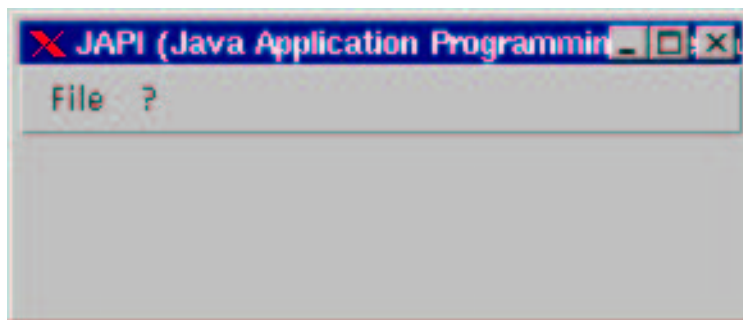


hasfocus
----------

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

## helpmenu

Synopsis	function <b>j_helpmenu</b> ( integer obj , sequence label )
Argumente	obj            integer label          sequence
Beschreibung	Die Funktion erzeugt einen Hilfe-Menueintrag mit der Beschriftung <b>label</b> in der Komponente <b>obj</b> , 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 <b>j_hide</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Zeigt die Komponente <b>obj</b> 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)
:

```

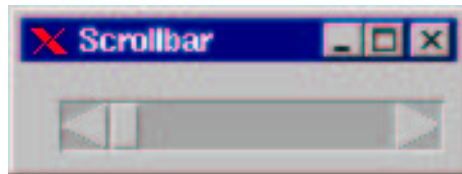


image
-------

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

inserttext
------------

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



isparent
----------

Synopsis	function <b>j_isparent</b> ( integer obj , integer cont )
Argumente	obj            integer cont           integer
Beschreibung	Liefert J.TRUE zurück, falls das <b>cont</b> die umgebende Komponente von <b>obj</b> 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 <b>j_isresizable</b> ( 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 <b>j_isselect</b> ( integer obj , integer item )
Argumente	obj            integer item           integer
Beschreibung	Liefert J_TRUE zurück, falls das Item mit dem Index <b>item</b> selektiert ist, sonst J_FALSE
Zielkomponenten	List

isvisible
-----------

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

keylistener
-------------

Synopsis	function <b>j_keylistener</b> ( integer obj )
Argumente	obj          integer
Beschreibung	Bindet an die Komponente <b>obj</b> 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 )

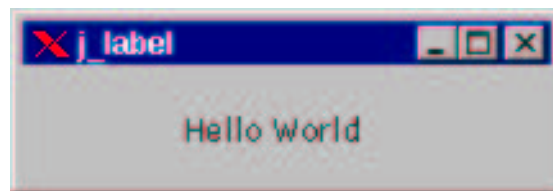
Argumente    obj            integer  
              label          sequence

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 <b>j_led</b> ( 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 <b>style = J_ROUND</b> oder eine eckige Form besitzen <b>style=J_RECT</b> . 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

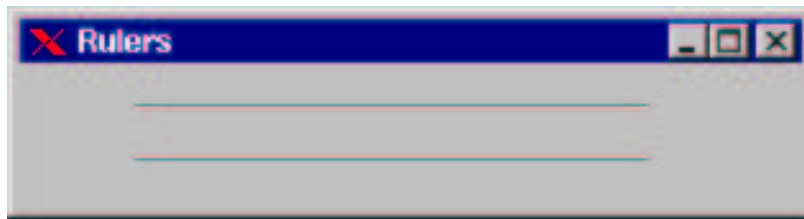
Beispiel

```

:
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 )`

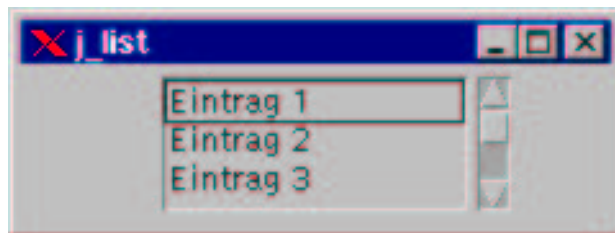
Argumente        `obj            integer`  
                   `rows          integer`

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

## menubar

Synopsis           function **j\_menubar** ( integer obj )

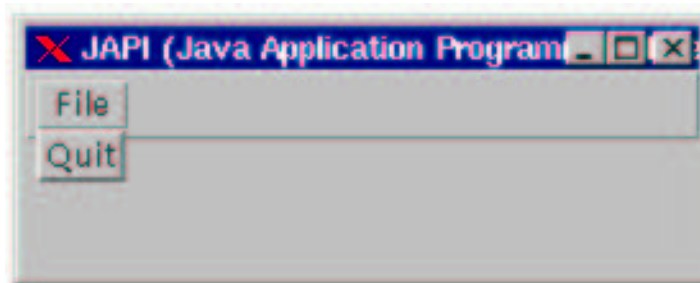
Argumente       obj           integer

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

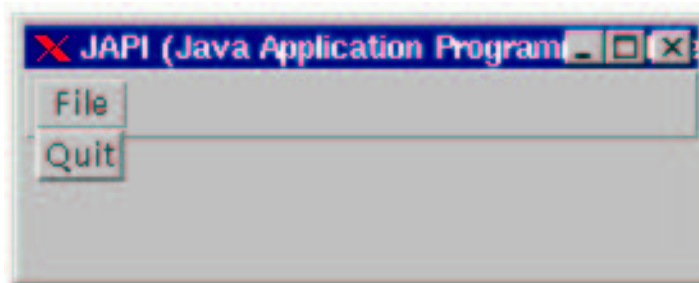
Synopsis	function <b>j_menuitem</b> ( integer obj , sequence label )
Argumente	obj            integer label          sequence
Beschreibung	Die Funktion erzeugt ein MenuItem mit der Beschriftung <b>label</b> in der Komponente <b>obj</b> , und liefert dessen Eventnummer zurück. Eine MenuItem erzeugt einen 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 )

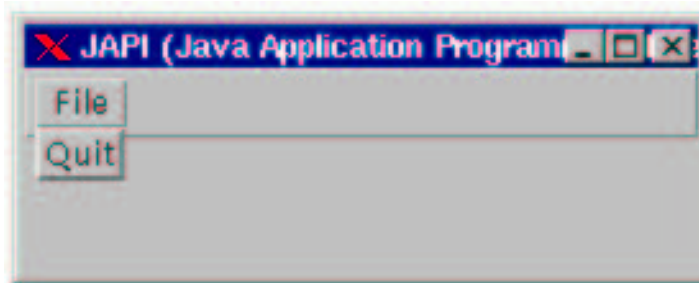
Argumente    obj            integer  
              str            sequence

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.

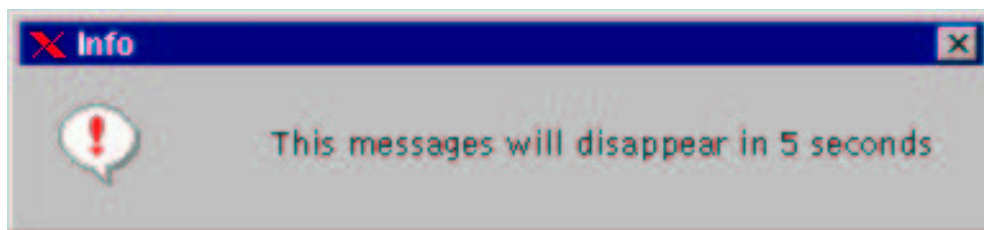
Zielkomponenten    Frame

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

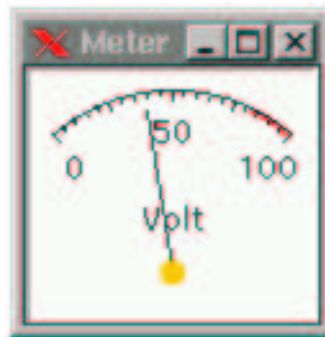
Zielkomponenten   Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```





mouselistener
---------------

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

multiplemode
--------------

Synopsis	function <b>j_multiplemode</b> ( integer obj , integer bool )
Argumente	obj            integer bool           integer
Beschreibung	die Komponente erlaubt mehrfache Selectierung, falls <b>bool</b> 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.

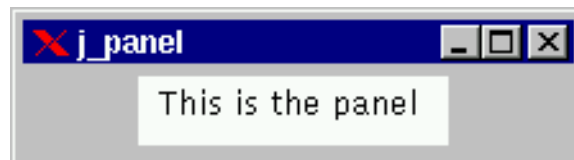
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

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
-----------

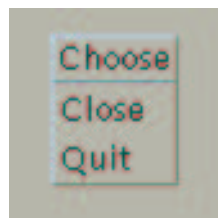
Synopsis	function <b>j_popupmenu</b> ( integer obj , sequence label )
Argumente	obj            integer label          sequence
Beschreibung	Diese Funktion erzeugt ein neues Popupmenu und liefert dessen Eventnummer zurück. Im Fehlerfall wird $-1$ zurückgegeben. Ein Popupmenu ist funktional identisch mit einem Menu, es ist allerdings an keine Menuleiste gebunden. Daher kann ein Popupmenu prinzipiell an jeder Position innerhalb der Oberfläche platziert werden. Wie an Menus kann auch an ein Popupmenu 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

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

print
-------

Synopsis	procedure <b>j_print</b> ( integer obj )
Argumente	obj            integer
Beschreibung	<p>Die Prozedur druckt den Inhalt der Komponente <b>obj</b>. 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.

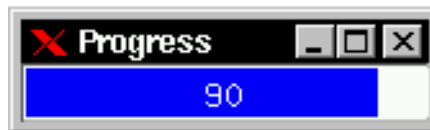
Zielkomponenten Panel, Borderpanel, Window, Dialog, Frame

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 <b>j_radiobutton</b> ( integer obj , sequence label )</code>				
Argumente	<table> <tr> <td style="padding-right: 20px;"><code>obj</code></td> <td><code>integer</code></td> </tr> <tr> <td><code>label</code></td> <td><code>sequence</code></td> </tr> </table>	<code>obj</code>	<code>integer</code>	<code>label</code>	<code>sequence</code>
<code>obj</code>	<code>integer</code>				
<code>label</code>	<code>sequence</code>				
Beschreibung	<p>Diese Funktion erzeugt einen Radiobutton mit dem Textinhalt <b>label</b> 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 <b>j_releaseall</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Gibt alle Elemente aus der Komponenten <b>obj</b> frei.
Zielkomponenten	Panel, Borderpanel, Window, Dialog, Frame

release
---------

Synopsis	procedure <b>j_release</b> ( integer obj )
Argumente	obj          integer
Beschreibung	Gibt die Komponente <b>obj</b> 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 <b>j_replacetext</b> ( integer obj , sequence text , integer start , integer end )
Argumente	obj           integer text           sequence start          integer end            integer
Beschreibung	Ersetzt den Text zwischen Position <b>start</b> bis <b>end</b> durch <b>text</b> .
Zielkomponenten	Textarea

saveimage

Synopsis	function <b>j_saveimage</b> ( 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 <b>filename</b> . Es werden zwei Fileformate unterstützt: <ul style="list-style-type: none"><li>• BMP</li><li>• PPM</li></ul>
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.

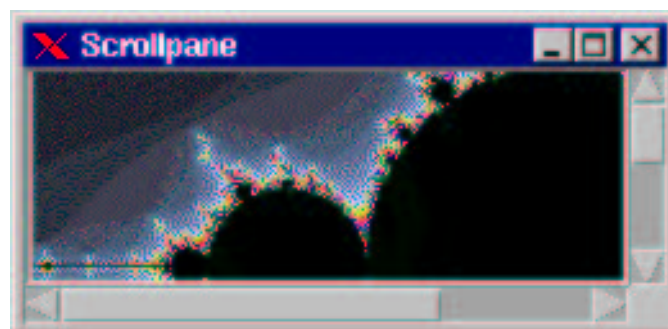
Zielkomponenten   Panel, Borderpanel, Window, Dialog, Frame

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 <b>j_selecttext</b> ( integer obj , integer start , integer end )
Argumente	obj            integer start          integer end            integer
Beschreibung	Selectiert den Text von Position <b>start</b> bis <b>end</b> .
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.

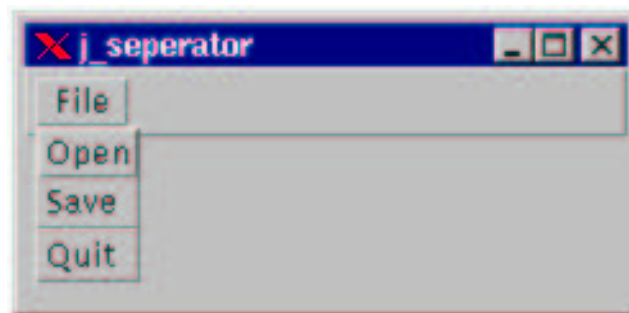
Zielkomponenten   Menu, HelpMenu, Popupmenu

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

setblockinc
-------------

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

setborderlayout
-----------------

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

setborderpos
--------------

Synopsis	procedure <b>j_setborderpos</b> ( integer obj , integer pos )
Argumente	obj            integer pos            integer
Beschreibung	Positioniert die Komponente <b>obj</b> an die angegebene Position <b>pos</b> . Im umgebenden Container muß einen BorderLayout Manager ein- gestellt 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, Graphiclabel, 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")
:

```



## setcolumns

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

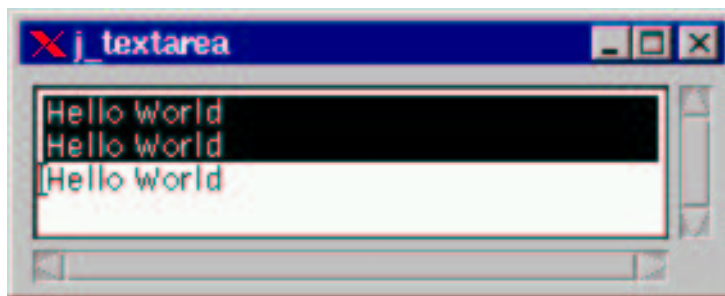
Argumente        `obj`            integer  
                  `columns`       integer

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 <b>j_setcursor</b> ( integer obj , integer cursor )
Argumente	obj            integer cursor        integer
Beschreibung	Setzt den Cursor der Komponenten <b>obj</b> auf <b>cursor</b> .
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     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:

- 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 <b>j_setechochar</b> ( integer obj , integer chr )
Argumente	obj            integer chr            integer
Beschreibung	Bestimmt das Zeichen <b>chr</b> als Echo Zeichen für alle Benutzereingaben in die Komponente .
Zielkomponenten	Textfield

seteditable
-------------

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

setfixlayout
--------------

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

setflowfill
-------------

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



setflowlayout
---------------

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

setfocus
----------

Synopsis	function <b>j_setfocus</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Weist den Input Focus der Komponenten <b>obj</b> 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 <b>j_setfontname</b> ( integer obj , integer name )
Argumente	obj            integer name          integer
Beschreibung	Diese Prozedur setzt den Fonttyp in der Komponenten <b>obj</b> , auf <b>name</b> . Der Parameter <b>name</b> kann folgende Werte annehmen: <ul style="list-style-type: none"> <li>• J_COURIER : Courier (Monospaced)</li> <li>• J_HELVETIA : Helvetia (Sans Serif)</li> <li>• J_TIMES : Times (Serif)</li> <li>• J_DIALOGIN : Dialog In</li> <li>• J_DIALOGOUT : Dialog Out</li> </ul>
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

```

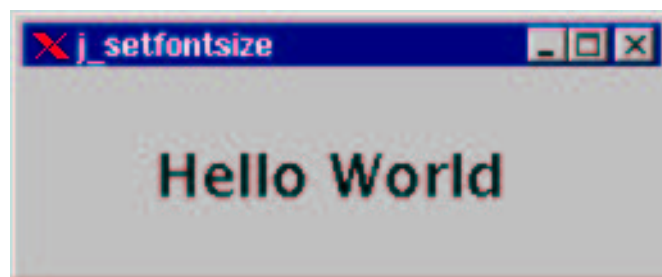
:
label = j_label(jframe,"Hello World")
j_setfontname(label,J_HELVETIA)
:

```



setfontsize
-------------

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



## setfontstyle

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

Beispiel

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



setgridlayout

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



## sethgap

Synopsis	procedure <b>j_sethgap</b> ( integer obj , integer hgap )
Argumente	obj           integer hgap           integer
Beschreibung	Setzt den horizontalen Abstand innerhalb der, durch <b>obj</b> referenzierten, Komponenten auf <b>hgap</b> 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 <b>hgap</b> 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.

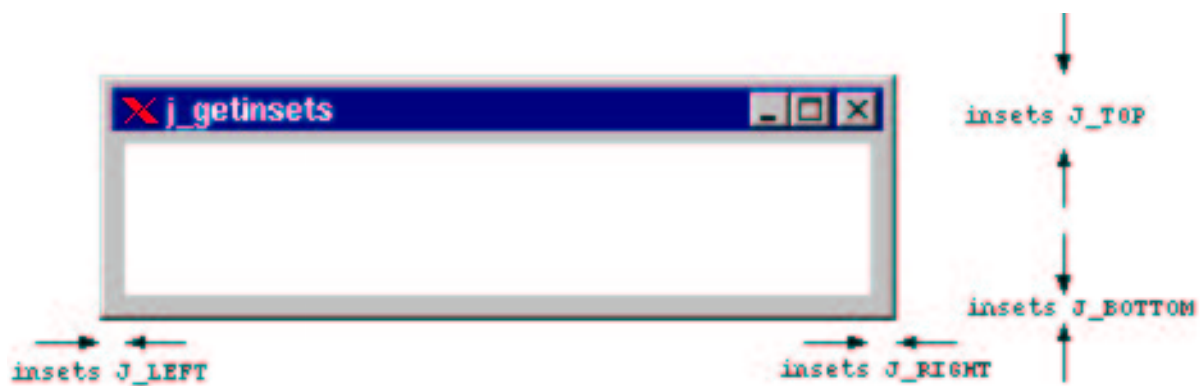
Zielkomponenten    Panel, Borderpanel, Window, Dialog, Frame

Beispiel

```

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

```



setmax
--------

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

setmin
--------

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

setnamedcolorbg
-----------------

Synopsis	procedure <b>j_setnamedcolorbg</b> ( integer obj , integer color )
Argumente	obj           integer color         integer
Beschreibung	Setzt die Hintergrundfarbe auf eine vordefinierte Farbe <b>color</b> .
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, Sevenssegment



setnolayout
-------------

Synopsis	procedure <b>j_setnolayout</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Entfernt den aktuellen Layout Manager aus der Komponenten <b>obj</b> .
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 <b>j_setpos</b> ( integer obj , integer xpos , integer ypos )
Argumente	obj           integer xpos           integer ypos           integer
Beschreibung	Verschiebt die Komponente <b>obj</b> zur angegebenen Koordinate ( <b>xpos,ypos</b> ) 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 <b>j_setradiogroup</b> ( integer rbutton, , integer rgroup )
Argumente	rbutton,     integer rgroup     integer
Beschreibung	Weist dem Radiobutton <b>rbutton</b> die Radiogroup <b>rgroup</b> 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 )

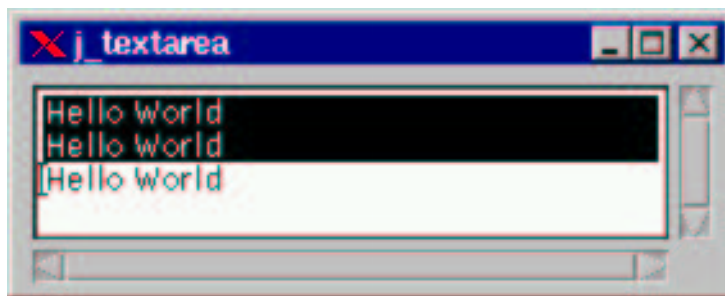
Argumente    obj            integer  
              rows          integer

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 <b>j_setshortcut</b> ( integer obj , integer chr )
Argumente	obj            integer chr            integer
Beschreibung	Bestimmt das Zeichen <b>chr</b> 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 <b>j_settext</b> ( integer obj , sequence str )
Argumente	obj            integer str            sequence
Beschreibung	Setzt den Textinhalt in der, durch <b>obj</b> referenzierten, Komponenten auf den Inhalt des Parameter <b>str</b> . 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 )

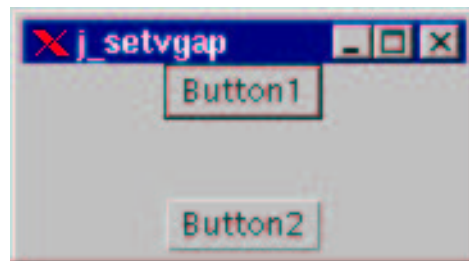
Argumente        obj            integer  
                  vgap           integer

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 <b>j_setxor</b> ( 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 <b>j_sevenssegment</b> ( 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 <b>j_showpopup</b> ( integer obj , integer xpos , integer ypos )
Argumente	obj           integer xpos           integer ypos           integer
Beschreibung	Zeigt die Komponente an der Koordinate ( <b>xpos,ypos</b> ) an.
Zielkomponenten	Popupmenu

show
------

Synopsis	procedure <b>j_show</b> ( integer obj )
Argumente	obj            integer
Beschreibung	Zeigt die Komponente <b>obj</b> 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 <b>j_sleep</b> ( integer msec )
Argumente	msec          integer
Beschreibung	Unterbricht die Ausführung der Applikation für <b>msec</b> 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

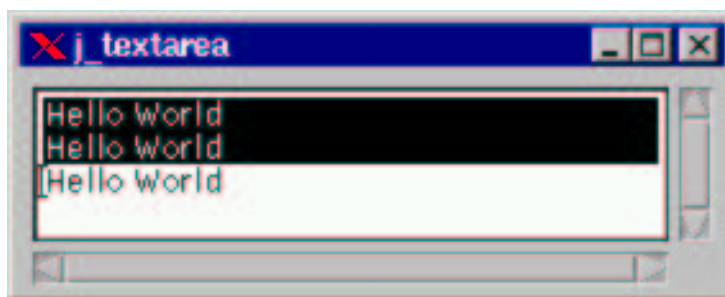
Beschreibung

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 *j\_setsize()* verwendet werden. Die Textarea besitzt voreingestellt zwei Scrollbars, die sich automatisch anpassen. Eine Textarea liefert einen Event, sobald sich der Textinhalt ändert.

Zielkomponenten    Panel, Borderpanel, Window, Dialog, Frame

Beispiel

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



## textfield

Synopsis	function <b>j_textfield</b> ( 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 <b>columns</b> Spalten, und ist somit abhängig vom voreingestellten Font. Soll eine Größe in Pixeleinheiten eingestellt werden, so kann nachträglich die Funktion <b>j_setsize()</b> verwendet werden. Ein Textfield liefert einen Event, sobald die Eingabe mit <b>Return</b> 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 <b>j_windowlistener</b> ( integer window , integer kind )
Argumente	window     integer kind        integer
Beschreibung	<p>Bindet an die Komponente <b>window</b> einen neuen Windowlistener, und liefert dessen Eventnummer zurück. Der Parameter <b>kind</b> bestimmt, wann ein Event ausgelöst wird. Zulässige Werte für <b>kind</b> sind:</p> <ul style="list-style-type: none"><li>• J_ACTIVATED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> aktiviert wird.</li><li>• J_DEACTIVATED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> deaktiviert wird.</li><li>• J_OPENED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> geöffnet ist.</li><li>• J_CLOSED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> geschlossen ist.</li><li>• J_ICONFIED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> iconifiziert wird.</li><li>• J_DEICONFIED : Der Componentlistener liefert einen Event, wenn die Komponente <b>obj</b> deiconifiziert wird.</li><li>• J_CLOSING : Der Componentlistener liefert einen Event, wenn das close icon der Komponenten <b>obj</b> angeklickt wird.</li></ul>
Zielkomponenten	Window, Dialog, Frame

## window

Synopsis	function <b>j_window</b> ( 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 <i>j_show()</i> 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