



Reference Manual

Merten Joost

Inhaltsverzeichnis

I	Reference	5
1	Components	7
	Button	7
	Borderpanel	10
	Canvas	15
	Checkbox	20
	CheckmenuItem	23
	Choice	24
	Dialog	27
	Focuslistener	32
	Frame	33
	Helpmenu	39
	Hscrollbar	40
	Graphicbutton	43
	Graphiclabel	46
	Image	49
	KeyListener	52
	Label	53
	List	56
	Menu	60
	MenuItem	62
	MouseListener	63
	Panel	64
	PopupMenu	69
	Printer	71
	Radiobutton	73
	ScrollPane	76
	Textarea	79
	Textfield	83
	Vscrollbar	87
	Window	90
2	Functions	95
	add	95
	additem	96

alertbox	97
appendtext	98
beep	99
borderpanel	100
button	101
canvas	102
checkbox	103
checkmenuItem	104
choice	105
choicebox2	106
choicebox3	107
cliprect	108
componentlistener	109
connect	110
delete	111
deselect	112
dialog	113
disable	114
dispose	115
drawarc	116
drawcircle	117
drawimage	118
drawimagesource	119
drawline	120
drawoval	121
drawpixel	122
drawpolygon	123
drawpolyline	124
drawrect	125
drawroundrect	126
drawscaledimage	127
drawstring	128
enable	129
filedialog	130
fileselector	131
fillarc	132
fillcircle	133
filloval	134
fillpolygon	135
fillrect	136
fillroundrect	137
focuslistener	138
frame	139
getaction	140
getcolumns	141
getcurpos	142
getfontascent	143

getfontheight	144
getheight	145
getimage	146
getimagesource	147
getinsets	148
getitem	149
getitemcount	150
getkeychar	151
getkeycode	152
getlayoutid	153
getlength	154
getmousebutton	155
getmousex	156
getmousey	157
getparent	158
getparentid	159
getrows	160
getscaledimage	161
getscreenheight	162
getscreenwidth	163
getselect	164
getselend	165
getselstart	166
getseltext	167
getstate	168
gettext	169
getvalue	170
getviewportheight	171
getviewportwidth	172
getwidth	173
getxpos	174
getypos	175
graphicbutton	176
graphiclabel	177
hasfocus	178
helpmenu	179
hide	180
hscrollbar	181
image	182
insert	183
inserttext	184
isparent	185
isresizable	186
isselect	187
isvisible	188
keylistener	189
label	190

line	191
list	192
loadimage	193
menu	194
menubar	195
menuitem	196
messagebox	197
mouselistener	198
multiplemode	199
nextaction	200
pack	201
panel	202
popupmenu	203
print	204
printer	205
quit	206
radiobutton	207
radiogroup	208
random	209
release	210
releaseall	211
remove	212
removeall	213
removeitem	214
replacetext	215
saveimage	216
scrollpane	217
select	218
selectall	219
selecttext	220
seperator	221
setalign	222
setblockinc	223
setborderlayout	224
setborderpos	225
setcolor	226
setcolorbg	227
setcolumns	228
setcurpos	229
setcursor	230
setdebug	231
setechochar	232
seteditable	233
setfixlayout	234
setflowfill	235
setflowlayout	236
setfocus	237

setFont	238
setFontname	239
setfontsize	240
setFontstyle	241
setgridlayout	242
sethgap	243
setIcon	244
setImage	245
setinsets	246
setMax	247
setmin	248
setnamedcolor	249
setnamedcolorbg	250
setnolayout	251
setpos	252
setradiogroup	253
setresizable	254
setrows	255
setshortcut	256
setsize	257
setslidesize	258
setstate	259
setText	260
setunitinc	261
setValue	262
setvgap	263
setxor	264
show	265
showpopup	266
sleep	267
start	268
sync	269
textarea	270
textfield	271
translate	272
vscrollbar	273
window	274
windowlistener	275

Teil I

Reference

Kapitel 1

Components

Button

j_button	<i>int j_button (int obj , char* label);</i> Creates a new button component with the specified label and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds button obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to button obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables button obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the button obj .
j_enable	<i>void j_enable (int obj);</i> enables the button obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to button obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of button obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of button obj .
j_getheight	<i>int j_getheight (int obj);</i>

	Returns the height of button obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of button 's label or text.
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the button 's text or label.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of button obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of button obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of button obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the button obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to button obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to button obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the button .
j_release	<i>void j_release (int obj);</i> Releases button obj from its parent component (container).

j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves button obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the button's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to button obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the button obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes button obj to specified width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the button obj to str .
j_show	<i>void j_show (int obj);</i> Shows the button obj .

Borderpanel

j_borderpanel	<i>int j_borderpanel (int obj , int type);</i> Creates a new borderpanel component with the style type and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds borderpanel obj to container cont
j_borderpanel	<i>int j_borderpanel (int obj , int type);</i> Creates a new borderpanel component with the style type and returns its event number.
j_button	<i>int j_button (int obj , char* label);</i> Creates a new button component with the specified label and returns its event number.
j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error -1 will be returned.
j_checkbox	<i>int j_checkbox (int obj , char* label);</i> Creates a new checkbox component with the specified label and returns its event number.
j_choice	<i>int j_choice (int obj);</i> Creates a new choice component and returns its event number.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to borderpanel obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables borderpanel obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the borderpanel obj .
j_enable	<i>void j_enable (int obj);</i> enables the borderpanel obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to borderpanel obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of borderpanel obj .

j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of borderpanel obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of borderpanel obj .
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Returns the width of the specified inset.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Returns the event number of the layoutmanager for containers obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of borderpanel obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of borderpanel obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of borderpanel obj in its parent's coordinate space.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>void j_hide (int obj);</i> Hides the borderpanel obj .
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to borderpanel obj , and returns its event number.

j_label	<i>int j_label (int obj , char* label);</i> Creates a new label component with the specified label and returns its event number.
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>int j_list (int obj , int rows);</i> Creates a new list component with the specified number of rows and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to borderpanel obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_pack	<i>void j_pack (int obj);</i> Resizes borderpanel to the minimal size of contained components.
j_panel	<i>int j_panel (int obj);</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the borderpanel .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Creates a new radiogroup and returns its event number.
j_release	<i>void j_release (int obj);</i> Releases borderpanel obj from its parent component (container).
j_releaseall	<i>void j_releaseall (int obj);</i> Releases all components from borderpanel obj .
j_scrollpane	<i>int j_scrollpane (int obj);</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>void j_setalign (int obj , int align);</i> Sets the alignment in borderpanel obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Adds a borderlayout manager to borderpanel obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves borderpanel obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.

j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the borderpanel 's obj cursor to the specified cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Adds a fixlayout manager to borderpanel obj (default layout manager).
j_setflowfill	<i>void j_setflowfill (int obj , int bool);</i> Resizes all containing component to the height (width) of borderpanel obj . Needs a flowlayout manager.
j_setflowlayout	<i>void j_setflowlayout (int obj , int align);</i> Adds a flowlayout manager to borderpanel obj with the specified alignment .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to borderpanel obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Adds a gridlayout manager to borderpanel obj with the specified rows and columns .
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Set the insets to the specified values.
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Removes the current layout manager from borderpanel obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the borderpanel obj to the specified Position (xpos,ypos).

j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes borderpanel obj to specified width and height .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Sets the vertical gap between components to hgap Pixel.
j_show	<i>void j_show (int obj);</i> Shows the borderpanel obj .
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.

Canvas

j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error -1 will be returned.
j_add	<i>void j_add (int obj , int cont);</i> Adds canvas obj to container cont
j_cliprect	<i>void j_cliprect (int obj , int x , int y , int width , int height);</i> Changes current clipping region to the specified rectangle (x , y , width , height).
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to canvas obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables canvas obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the canvas obj .
j_drawarc	<i>void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> Draws an unfilled arc from angle arc1 to angle arc2 with the center (x , y) and the horizontal radius rx and the vertical radius ry .
j_drawcircle	<i>void j_drawcircle (int obj , int x , int y , int r);</i> Draws an unfilled circle with center (x , y) and radius x .
j_drawimage	<i>void j_drawimage (int obj , int image , int x , int y);</i> Copies the image, given by its eventnumber image , to position (x , y).
j_drawimagesource	<i>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> Paints an image at Position (x , y) with width and height . The red, green and blue values of each pixel are given by the arrays r , g , b .
j_drawline	<i>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</i> Draws a line connecting (x1 , y1) and (x2 , y2).
j_drawoval	<i>void j_drawoval (int obj , int x , int y , int rx , int ry);</i> Draws an unfilled oval with the center (x , y) and the horizontal radius rx and the vertical radius ry .
j_drawpixel	<i>void j_drawpixel (int obj , int x , int y);</i> Draws a pixel at (x , y).

j_drawpolygon	<i>void j_drawpolygon (int obj , int len , int* x , int* y);</i> Draws an unfilled polygon based on first len elements in x and y .
j_drawpolyline	<i>void j_drawpolyline (int obj , int len , int* x , int* y);</i> Draws a series of line segments based on first len elements in x and y .
j_drawrect	<i>void j_drawrect (int obj , int x , int y , int width , int height);</i> Draws an unfilled rectangle from (x,y) of size width x height .
j_drawroundrect	<i>void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);</i> Draws an unfilled rectangle from (x,y) of size width x height with rounded corners. arcx and arcy specify the radius of rectangle corners.
j_drawscaledimage	<i>void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);</i> Copy the contents of the rectangular area defined by x , y , width sw , and height sh of the image to position (tx, ty) . The area will be scaled to target width th and target height th .
j_drawstring	<i>void j_drawstring (int obj , int x , int y , char* str);</i> Draws text on screen at position (x,y) .
j_enable	<i>void j_enable (int obj);</i> enables the canvas obj .
j_fillarc	<i>void j_fillarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> Draws an filled arc from angle arc1 to angle arc2 with the center (x, y) and the horizontal radius rx and the vertical radius ry .
j_fillcircle	<i>void j_fillcircle (int obj , int x , int y , int r);</i> Draws an filled circle with center (x, y) and radius x .
j_filloval	<i>void j_filloval (int obj , int x , int y , int rx , int ry);</i> Draws an filled oval with the center (x, y) and the horizontal radius rx and the vertical radius ry .
j_fillpolygon	<i>void j_fillpolygon (int obj , int len , int* x , int* y);</i> Draws an filled polygon based on first len elements in x and y .
j_fillrect	<i>void j_fillrect (int obj , int x , int y , int width , int height);</i> Draws an filled rectangle from (x,y) of size width x height .
j_fillroundrect	<i>void j_fillroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);</i> Draws an filled rectangle from (x,y) of size width x height with rounded corners. arcx and arcy specify the radius of rectangle corners.
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to canvas obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i>

	Returns the ascent (space above the baseline) of the actual font of canvas obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of canvas obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of canvas obj .
j_getimage	<i>int j_getimage (int obj);</i> Copy the contents of canvas obj into an image and return its eventnumber.
j_getimagesource	<i>int j_getimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> Returns an image of the specified size (x , y , width , height) of canvas . The red, green and blue values of each pixel will be stored in r , g , b
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getscaledimage	<i>int j_getscaledimage (int obj , int x , int y , int sw , int sh , int tw , int th);</i> Copy the contents of the rectangular area defined by x , y , width sw , and height sh into an image and return its eventnumber. The image will be scaled to target width th and target height th .
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of canvas obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of canvas obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of canvas obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the canvas obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to canvas obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i>

	<p>Adds a new mouse listener to canvas obj, and returns its event number. An event occurs, if the user action is of kind kind.</p>
j_popupmenu	<p><i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.</p>
j_print	<p><i>void j_print (int obj);</i> prints the canvas .</p>
j_release	<p><i>void j_release (int obj);</i> Releases canvas obj from its parent component (container).</p>
j_setborderpos	<p><i>void j_setborderpos (int obj , int pos);</i> Moves canvas obj at a certain position. The outer container needs a border layout manager.</p>
j_setcolor	<p><i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r, g, b) values.</p>
j_setcolorbg	<p><i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r, g, b) values.</p>
j_setcursor	<p><i>int j_setcursor (int obj , int cursor);</i> Changes the canvas 's obj cursor to the specified cursor.</p>
j_setfocus	<p><i>int j_setfocus (int obj);</i> Directs the input focus to canvas obj.</p>
j_setfont	<p><i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name, style and size.</p>
j_setfontname	<p><i>void j_setfontname (int obj , int name);</i> Changes the font to the given name.</p>
j_setfontsize	<p><i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size.</p>
j_setfontstyle	<p><i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style.</p>
j_setnamedcolor	<p><i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color.</p>
j_setnamedcolorbg	<p><i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color.</p>
j_setpos	<p><i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the canvas obj to the specified Position (xpos,ypos).</p>
j_setsize	<p><i>void j_setsize (int obj , int width , int height);</i> Resizes canvas obj to specified width and height.</p>

j_setxor	<i>void j_setxor (int obj , int bool);</i> Changes painting mode to XOR mode, if bool = J_TRUE . In this mode, drawing the same object in the same color at the same location twice has no net effect.
j_show	<i>void j_show (int obj);</i> Shows the canvas obj .
j_translate	<i>void j_translate (int obj , int x , int y);</i> Moves the origin of drawing operations to (x , y).

Checkbox

j_checkbox	<i>int j_checkbox (int obj , char* label);</i> Creates a new checkbox component with the specified label and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds checkbox obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to checkbox obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables checkbox obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the checkbox obj .
j_enable	<i>void j_enable (int obj);</i> enables the checkbox obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to checkbox obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of checkbox obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of checkbox obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of checkbox obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstate	<i>int j_getstate (int obj);</i> Returns J_TRUE , if checkbox is selected, J_FALSE otherwise.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the checkbox 's text or label.

j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of checkbox obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of checkbox obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of checkbox obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the checkbox obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to checkbox obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to checkbox obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the checkbox .
j_release	<i>void j_release (int obj);</i> Releases checkbox obj from its parent component (container).
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves checkbox obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the checkbox 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to checkbox obj .

j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the checkbox obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes checkbox obj to specified width and height .
j_setstate	<i>void j_setstate (int obj , int bool);</i> The checkbox becomes selected, if bool is J_TRUE .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the checkbox obj to str .
j_show	<i>void j_show (int obj);</i> Shows the checkbox obj .

Checkmenuitem

j_checkmenuitem	<i>int j_checkmenuitem (int obj , char* label);</i> creates a new checkmenuitem with the specified label and returns its event number.
j_disable	<i>void j_disable (int obj);</i> Disables checkmenuitem obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the checkmenuitem obj .
j_enable	<i>void j_enable (int obj);</i> enables the checkmenuitem obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of checkmenuitem 's label or text.
j_getstate	<i>int j_getstate (int obj);</i> Returns J_TRUE , if checkmenuitem is selected, J_FALSE otherwise.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the checkmenuitem 's text or label.
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setshortcut	<i>void j_setshortcut (int obj , char chr);</i> Changes the shortcut chr of the checkmenuitem .
j_setstate	<i>void j_setstate (int obj , int bool);</i> The checkmenuitem becomes selected, if bool is J_TRUE .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the checkmenuitem obj to str .

Choice

j_choice	<i>int j_choice (int obj);</i> Creates a new choice component and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds choice obj to container cont
j_additem	<i>void j_additem (int obj , char* str);</i> adds a new item containing str to choice obj .
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to choice obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables choice obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the choice obj .
j_enable	<i>void j_enable (int obj);</i> enables the choice obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to choice obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of choice obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of choice obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of choice obj .
j_getitem	<i>char* j_getitem (int obj , int item , char* str);</i> returns the label of the given item .
j_getitemcount	<i>int j_getitemcount (int obj);</i> Returns the number of items of choice obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i>

	Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getselect	<i>int j_getselect (int obj);</i> Returns the position of currently selected item.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of choice obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of choice obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of choice obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the choice obj .
j_insert	<i>int j_insert (int obj , int pos , char* label);</i> inserts a new item to choice obj at position pos with the specified label .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to choice obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to choice obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the choice .
j_release	<i>void j_release (int obj);</i> Releases choice obj from its parent component (container).
j_remove	<i>int j_remove (int obj , int item);</i> removes the Item with the Index item from the choice .
j_removeall	<i>int j_removeall (int obj);</i> Removes all items from the choice .
j_removeitem	<i>int j_removeitem (int obj , char* item);</i>

remove the first occurrence of **item** from the choice .

j_select	<i>int j_select (int obj , int item);</i> Makes the given item the selected one for the choice .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves choice obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the choice 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to choice obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the choice obj to the specified Position (xpos , ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes choice obj to specified width and height .
j_show	<i>void j_show (int obj);</i> Shows the choice obj .

Dialog

j_dialog	<i>int j_dialog (int obj , char* label);</i> Creates a new dialog window with the specified label and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds dialog obj to container cont
j_borderpanel	<i>int j_borderpanel (int obj , int type);</i> Creates a new borderpanel component with the style type and returns its event number.
j_button	<i>int j_button (int obj , char* label);</i> Creates a new button component with the specified label and returns its event number.
j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error -1 will be returned.
j_checkbox	<i>int j_checkbox (int obj , char* label);</i> Creates a new checkbox component with the specified label and returns its event number.
j_choice	<i>int j_choice (int obj);</i> Creates a new choice component and returns its event number.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to dialog obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables dialog obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the dialog obj .
j_enable	<i>void j_enable (int obj);</i> enables the dialog obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to dialog obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of dialog obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i>

	Returns the total pixel height of the actual font of dialog obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of dialog obj .
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Returns the width of the specified inset.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Returns the event number of the layoutmanager for containers obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of dialog 's label or text.
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the dialog 's text or label.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of dialog obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of dialog obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of dialog obj in its parent's coordinate space.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>void j_hide (int obj);</i> Hides the dialog obj .
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.

j_isresizable	<i>int j_isresizable (int obj);</i> returns true if dialog is resizable, false otherwise
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to dialog obj , and returns its event number.
j_label	<i>int j_label (int obj , char* label);</i> Creates a new label component with the specified label and returns its event number.
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>int j_list (int obj , int rows);</i> Creates a new list component with the specified number of rows and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to dialog obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_pack	<i>void j_pack (int obj);</i> Resizes dialog to the minimal size of contained components.
j_panel	<i>int j_panel (int obj);</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the dialog .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Creates a new radiogroup and returns its event number.
j_release	<i>void j_release (int obj);</i> Releases dialog obj from its parent component (container).
j_releaseall	<i>void j_releaseall (int obj);</i> Releases all components from dialog obj .
j_scrollpane	<i>int j_scrollpane (int obj);</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>void j_setalign (int obj , int align);</i> Sets the alignment in dialog obj to align . Needs a flowlayout Manager.

j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Adds a borderlayout manager to dialog obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves dialog obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the dialog 's obj cursor to the specified cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Adds a fixlayout manager to dialog obj (default layout manager).
j_setflowfill	<i>void j_setflowfill (int obj , int bool);</i> Resizes all containing component to the height (width) of dialog obj . Needs a flowlayout manager.
j_setflowlayout	<i>void j_setflowlayout (int obj , int align);</i> Adds a flowlayout manager to dialog obj with the specified alignment .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to dialog obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Adds a gridlayout manager to dialog obj with the specified rows and columns .
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Set the insets to the specified values.
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .

j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Removes the current layout manager from dialog obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the dialog obj to the specified Position (xpos,ypos).
j_setresizable	<i>void j_setresizable (int obj , int resizable);</i> The dialog cannot be resized, if resizable is J_FALSE .
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes dialog obj to specified width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the dialog obj to str .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Sets the vertical gap between components to vgap Pixel.
j_show	<i>void j_show (int obj);</i> Shows the dialog obj .
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.
j_windowlistener	<i>int j_windowlistener (int window , int kind);</i> Adds a new windowlistener to obj , and returns its event number. An event occurs, if the user action is of kind kind .

Focuslistener

j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to focuslistener obj , and returns its event number.
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the focuslistener obj .
j_hasfocus	<i>int j_hasfocus (int obj);</i> Returns J_TRUE if the focuslistener has the focus, J_FALSE otherwise.

Frame

j_frame	<i>int j_frame (char* label);</i> Creates a new frame component with the specified label and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds frame obj to container cont
j_alertbox	<i>void j_alertbox (int obj , char* title , char* text , char* button);</i> Shows a alertbox with the specified title , text and button .
j_borderpanel	<i>int j_borderpanel (int obj , int type);</i> Creates a new borderpanel component with the style type and returns its event number.
j_button	<i>int j_button (int obj , char* label);</i> Creates a new button component with the specified label and returns its event number.
j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error -1 will be returned.
j_checkbox	<i>int j_checkbox (int obj , char* label);</i> Creates a new checkbox component with the specified label and returns its event number.
j_choice	<i>int j_choice (int obj);</i> Creates a new choice component and returns its event number.
j_choicebox2	<i>void j_choicebox2 (int obj , char* title , char* text , char* button1 , char* button2);</i> Shows a choicebox with the specified title , text and two buttons.
j_choicebox3	<i>void j_choicebox3 (int obj , char* title , char* text , char* button1 , char* button2 , char* button3);</i> Shows a choicebox with the specified title , text and three buttons.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to frame obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_dialog	<i>int j_dialog (int obj , char* label);</i> Creates a new dialog window with the specified label and returns its event number.
j_disable	<i>void j_disable (int obj);</i>

	Disables frame obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the frame obj .
j_enable	<i>void j_enable (int obj);</i> enables the frame obj .
j_filedialog	<i>char* j_filedialog (int frame , char* title , char* directory , char* filename);</i> Opens a filedialog box in the specified directory with the specified title and returns the selected filename .
j_fileselector	<i>char* j_fileselector (int frame , char* title , char* filter , char* filename);</i> Opens a fileselector box with the preselected filename and the specified title and returns the selected filename .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to frame obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of frame obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of frame obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of frame obj .
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Returns the width of the specified inset.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Returns the event number of the layoutmanager for containers obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of frame 's label or text.
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the frame 's text or label.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of frame obj .
j_getxpos	<i>int j_getxpos (int obj);</i>

	Returns the current horizontal position of frame obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of frame obj in its parent's coordinate space.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>void j_hide (int obj);</i> Hides the frame obj .
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isresizable	<i>int j_isresizable (int obj);</i> returns true if frame is resizable, false otherwise
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to frame obj , and returns its event number.
j_label	<i>int j_label (int obj , char* label);</i> Creates a new label component with the specified label and returns its event number.
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>int j_list (int obj , int rows);</i> Creates a new list component with the specified number of rows and returns its event number.
j_menubar	<i>int j_menubar (int obj);</i> Creates a new menubar and returns its event number.
j_messagebox	<i>void j_messagebox (int obj , char* title , char* text);</i> Shows a messagebox with the specified title and text and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i>

Adds a new mouse listener to frame **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.

j_pack	<i>void j_pack (int obj);</i> Resizes frame to the minimal size of contained components.
j_panel	<i>int j_panel (int obj);</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the frame .
j_printer	<i>int j_printer (int frame);</i> Creates a new object, representing a paper of the printer.
j_radiogroup	<i>int j_radiogroup (int obj);</i> Creates a new radiogroup and returns its event number.
j_release	<i>void j_release (int obj);</i> Releases frame obj from its parent component (container).
j_releaseall	<i>void j_releaseall (int obj);</i> Releases all components from frame obj .
j_scrollpane	<i>int j_scrollpane (int obj);</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>void j_setalign (int obj , int align);</i> Sets the alignment in frame obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Adds a borderlayout manager to frame obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves frame obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the frame 's obj cursor to the specified cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Adds a fixlayout manager to frame obj (default layout manager).

j_setflowfill	<i>void j_setflowfill (int obj , int bool);</i> Resizes all containing component to the height (width) of frame obj . Needs a flowlayout manager.
j_setflowlayout	<i>void j_setflowlayout (int obj , int align);</i> Adds a flowlayout manager to frame obj with the specified alignment .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to frame obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Adds a gridlayout manager to frame obj with the specified rows and columns .
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Sets the horizontal gap between components to hgap Pixel.
j_seticon	<i>void j_seticon (int frame , int icon);</i> Sets the image icon to display when the frame is iconized. Not all platforms support the concept of iconizing a window.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Set the insets to the specified values.
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Removes the current layout manager from frame obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the frame obj to the specified Position (xpos,ypos).
j_setresizable	<i>void j_setresizable (int obj , int resizable);</i> The frame cannot be resized, if resizable is J_FALSE .
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes frame obj to specified width and height .

j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the frame obj to str .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Sets the vertical gap between components to hgap Pixel.
j_show	<i>void j_show (int obj);</i> Shows the frame obj .
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.
j_window	<i>int j_window (int obj);</i> Creates a new simple window and returns its event number.
j_windowlistener	<i>int j_windowlistener (int window , int kind);</i> Adds a new windowlistener to obj , and returns its event number. An event occurs, if the user action is of kind kind .

Helpmenu

j_helpmenu	<i>int j_helpmenu (int obj , char* label);</i> Creates a new helpmenu component with the specified label and returns its event number.
j_checkmenuitem	<i>int j_checkmenuitem (int obj , char* label);</i> creates a new checkmenuitem with the specified label and returns its event number.
j_disable	<i>void j_disable (int obj);</i> Disables helpmenu obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the helpmenu obj .
j_enable	<i>void j_enable (int obj);</i> enables the helpmenu obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of helpmenu 's label or text.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the helpmenu 's text or label.
j_menuitem	<i>int j_menuitem (int obj , char* label);</i> Creates a new menuitem with the specified label and returns its event number.
j_seperator	<i>void j_seperator (int obj);</i> Adds a separator bar to the helpmenu .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setshortcut	<i>void j_setshortcut (int obj , char chr);</i> Changes the shortcut chr of the helpmenu .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the helpmenu obj to str .

Hscrollbar

j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds hscrollbar obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to hscrollbar obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables hscrollbar obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the hscrollbar obj .
j_enable	<i>void j_enable (int obj);</i> enables the hscrollbar obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to hscrollbar obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of hscrollbar obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of hscrollbar obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of hscrollbar obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getvalue	<i>int j_getvalue (int obj);</i> Returns the current setting of the scrollbar.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of hscrollbar obj .

j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of hscrollbar obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of hscrollbar obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the hscrollbar obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to hscrollbar obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to hscrollbar obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the hscrollbar .
j_release	<i>void j_release (int obj);</i> Releases hscrollbar obj from its parent component (container).
j_setblockinc	<i>int j_setblockinc (int obj , int val);</i> Changes the block increment amount for the hscrollbar to val .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves hscrollbar obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the hscrollbar 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to hscrollbar obj .

j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setmax	<i>int j_setmax (int obj , int val);</i> Changes the maximum value for the hscrollbar to val .
j_setmin	<i>int j_setmin (int obj , int val);</i> Changes the minimum value for the hscrollbar to val .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the hscrollbar obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes hscrollbar obj to specified width and height .
j_setslidesize	<i>int j_setslidesize (int obj , int val);</i> Changes the slide size to val .
j_setunitinc	<i>int j_setunitinc (int obj , int val);</i> Changes the unit increment amount for the hscrollbar to val .
j_setvalue	<i>void j_setvalue (int obj , int val);</i> Changes the current value of the hscrollbar to val .
j_show	<i>void j_show (int obj);</i> Shows the hscrollbar obj .

Graphicbutton

j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds graphicbutton obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to graphicbutton obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables graphicbutton obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the graphicbutton obj .
j_enable	<i>void j_enable (int obj);</i> enables the graphicbutton obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to graphicbutton obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of graphicbutton obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of graphicbutton obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of graphicbutton obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of graphicbutton obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of graphicbutton obj in its parent's coordinate space.

j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of graphicbutton obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the graphicbutton obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to graphicbutton obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to graphicbutton obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the graphicbutton .
j_release	<i>void j_release (int obj);</i> Releases graphicbutton obj from its parent component (container).
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves graphicbutton obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the graphicbutton's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to graphicbutton obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .

j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setimage	<i>void j_setimage (int obj , int image);</i> Sets the image to be displayed in obj .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the graphicbutton obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes graphicbutton obj to specified width and height .
j_show	<i>void j_show (int obj);</i> Shows the graphicbutton obj .

Graphiclabel

j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds graphiclabel obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to graphiclabel obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables graphiclabel obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the graphiclabel obj .
j_enable	<i>void j_enable (int obj);</i> enables the graphiclabel obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to graphiclabel obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of graphiclabel obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of graphiclabel obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of graphiclabel obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of graphiclabel obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of graphiclabel obj in its parent's coordinate space.

j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of graphiclabel obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the graphiclabel obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to graphiclabel obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to graphiclabel obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the graphiclabel .
j_release	<i>void j_release (int obj);</i> Releases graphiclabel obj from its parent component (container).
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves graphiclabel obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the graphiclabel 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to graphiclabel obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .

j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setimage	<i>void j_setimage (int obj , int image);</i> Sets the image to be displayed in obj .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the graphiclabel obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes graphiclabel obj to specified width and height .
j_show	<i>void j_show (int obj);</i> Shows the graphiclabel obj .

Image

j_image	<i>int j_image (int width , int height);</i> Creates a new (memory) image component with the given width and height and returns its event number.
j_cliprect	<i>void j_cliprect (int obj , int x , int y , int width , int height);</i> Changes current clipping region to the specified rectangle (x , y , width , height).
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the image obj .
j_drawarc	<i>void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> Draws an unfilled arc from angle arc1 to angle arc2 with the center (x , y) and the horizontal radius rx and the vertical radius ry .
j_drawcircle	<i>void j_drawcircle (int obj , int x , int y , int r);</i> Draws an unfilled circle with center (x , y) and radius x .
j_drawimage	<i>void j_drawimage (int obj , int image , int x , int y);</i> Copies the image, given by its eventnumber image , to position (x , y).
j_drawimagesource	<i>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> Paints an image at Position (x , y) with width and height . The red, green and blue values of each pixel are given by the arrays r , g , b .
j_drawline	<i>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</i> Draws a line connecting (x1 , y1) and (x2 , y2).
j_drawoval	<i>void j_drawoval (int obj , int x , int y , int rx , int ry);</i> Draws an unfilled oval with the center (x , y) and the horizontal radius rx and the vertical radius ry .
j_drawpixel	<i>void j_drawpixel (int obj , int x , int y);</i> Draws a pixel at (x , y).
j_drawpolygon	<i>void j_drawpolygon (int obj , int len , int* x , int* y);</i> Draws an unfilled polygon based on first len elements in x and y .
j_drawpolyline	<i>void j_drawpolyline (int obj , int len , int* x , int* y);</i> Draws a series of line segments based on first len elements in x and y .
j_drawrect	<i>void j_drawrect (int obj , int x , int y , int width , int height);</i> Draws an unfilled rectangle from (x , y) of size width x height .
j_drawroundrect	<i>void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);</i>

Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

- j_drawscaledimage** *void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);*
Copy the contents of the rectangular area defined by **x, y,** width **sw,** and height **sh** of the **image** to position **(tx, ty.** The area will be scaled to target width **th** and target height **th.**
- j_drawstring** *void j_drawstring (int obj , int x , int y , char* str);*
Draws text on screen at position **(x,y).**
- j_fillarc** *void j_fillarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);*
Draws an filled arc from angle **arc1** to angle **arc2** with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry.**
- j_fillcircle** *void j_fillcircle (int obj , int x , int y , int r);*
Draws an filled circle with center **(x, y)** and radius **x.**
- j_filloval** *void j_filloval (int obj , int x , int y , int rx , int ry);*
Draws an filled oval with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry.**
- j_fillpolygon** *void j_fillpolygon (int obj , int len , int* x , int* y);*
Draws an filled polygon based on first **len** elements in **x** and **y.**
- j_fillrect** *void j_fillrect (int obj , int x , int y , int width , int height);*
Draws an filled rectangle from **(x,y)** of size **width** x **height.**
- j_fillroundrect** *void j_fillroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);*
Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j_getheight** *int j_getheight (int obj);*
Returns the height of image **obj.**
- j_getimage** *int j_getimage (int obj);*
Copy the contents of image **obj** into an image and return its eventnumber.
- j_getimagesource** *int j_getimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);*
Returns an image of the specified size **(x, y, width, height)** of image . The red, green and blue values of each pixel will be stored in **r, g, b**
- j_getscaledimage** *int j_getscaledimage (int obj , int x , int y , int sw , int sh , int tw , int th);*
Copy the contents of the rectangular area defined by **x, y,** width **sw,** and height **sh** into an image and return its eventnumber. The image will be scaled to target width **th** and target height **th.**
- j_getwidth** *int j_getwidth (int obj);*
Returns the width of image **obj.**

j_print	<i>void j_print (int obj);</i> prints the image .
j_setxor	<i>void j_setxor (int obj , int bool);</i> Changes painting mode to XOR mode, if bool = J_TRUE . In this mode, drawing the same object in the same color at the same location twice has no net effect.
j_translate	<i>void j_translate (int obj , int x , int y);</i> Moves the origin of drawing operations to (x , y).

KeyListenerer

j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to keylistener obj , and returns its event number.
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the keylistener obj .
j_getkeychar	<i>int j_getkeychar (int obj);</i> Returns the ascii value of the last pressed key.
j_getkeycode	<i>int j_getkeycode (int obj);</i> Returns the integer key code of the last pressed key.

Label

j_label	<i>int j_label (int obj , char* label);</i> Creates a new label component with the specified label and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds label obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to label obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables label obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the label obj .
j_enable	<i>void j_enable (int obj);</i> enables the label obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to label obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of label obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of label obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of label obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the label 's text or label.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of label obj .
j_getxpos	<i>int j_getxpos (int obj);</i>

	Returns the current horizontal position of label obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of label obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the label obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to label obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to label obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the label .
j_release	<i>void j_release (int obj);</i> Releases label obj from its parent component (container).
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves label obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the label's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to label obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i>

	Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the label obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes label obj to specified width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the label obj to str .
j_show	<i>void j_show (int obj);</i> Shows the label obj .

List

j_list	<i>int j_list (int obj , int rows);</i> Creates a new list component with the specified number of rows and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds list obj to container cont
j_additem	<i>void j_additem (int obj , char* str);</i> adds a new item containing str to list obj .
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to list obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_deselect	<i>int j_deselect (int obj , int item);</i> Deselects the item at the designated position item , if selected.
j_disable	<i>void j_disable (int obj);</i> Disables list obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the list obj .
j_enable	<i>void j_enable (int obj);</i> enables the list obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to list obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of list obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of list obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of list obj .
j_getitem	<i>char* j_getitem (int obj , int item , char* str);</i> returns the label of the given item .
j_getitemcount	<i>int j_getitemcount (int obj);</i> Returns the number of items of list obj .
j_getparent	<i>int j_getparent (int obj);</i>

	Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getselect	<i>int j_getselect (int obj);</i> Returns the position of currently selected item.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of list obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of list obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of list obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the list obj .
j_insert	<i>int j_insert (int obj , int pos , char* label);</i> inserts a new item to list obj at position pos with the specified label .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_iselect	<i>int j_iselect (int obj , int item);</i> Returns J_TRUE if the particular item is currently selected, J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to list obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to list obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_multiplemode	<i>int j_multiplemode (int obj , int bool);</i> if bool is J_TRUE , selection mode is turned to multiplemode.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the list .

j_release	<i>void j_release (int obj);</i> Releases list obj from its parent component (container).
j_remove	<i>int j_remove (int obj , int item);</i> removes the Item with the Index item from the list .
j_removeall	<i>int j_removeall (int obj);</i> Removes all items from the list .
j_removeitem	<i>int j_removeitem (int obj , char* item);</i> remove the first occurrence of item from the list .
j_select	<i>int j_select (int obj , int item);</i> Makes the given item the selected one for the list .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves list obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the list 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to list obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the list obj to the specified Position (xpos,ypos).

j_setsize *void j_setsize (int obj , int width , int height);*
Resizes list **obj** to specified **width** and **height**.

j_show *void j_show (int obj);*
Shows the list **obj**.

Menu

j_menu	<i>int j_menu (int obj , char* str);</i> Creates a new menu component with the specified label and returns its event number.
j_checkmenuitem	<i>int j_checkmenuitem (int obj , char* label);</i> creates a new checkmenuitem with the specified label and returns its event number.
j_disable	<i>void j_disable (int obj);</i> Disables menu obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the menu obj .
j_enable	<i>void j_enable (int obj);</i> enables the menu obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of menu 's label or text.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the menu 's text or label.
j_helpmenu	<i>int j_helpmenu (int obj , char* label);</i> Creates a new helpmenu component with the specified label and returns its event number.
j_menu	<i>int j_menu (int obj , char* str);</i> Creates a new menu component with the specified label and returns its event number.
j_menuitem	<i>int j_menuitem (int obj , char* label);</i> Creates a new menuitem with the specified label and returns its event number.
j_seperator	<i>void j_seperator (int obj);</i> Adds a separator bar to the menu .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i>

	Changes the font to the given style .
j_setshortcut	<i>void j_setshortcut (int obj , char chr);</i> Changes the shortcut chr of the menu .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the menu obj to str .

Menuitem

j_menuitem	<i>int j_menuitem (int obj , char* label);</i> Creates a new menuitem with the specified label and returns its event number.
j_disable	<i>void j_disable (int obj);</i> Disables menuitem obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the menuitem obj .
j_enable	<i>void j_enable (int obj);</i> enables the menuitem obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of menuitem 's label or text.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the menuitem 's text or label.
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setshortcut	<i>void j_setshortcut (int obj , char chr);</i> Changes the shortcut chr of the menuitem .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the menuitem obj to str .

Mouselistener

j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to mouselistener obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the mouselistener obj .
j_getmousebutton	<i>int j_getmousebutton (int mouselistener);</i> Returns the latest used mousebutton.
j_getmousex	<i>int j_getmousex (int mouselistener);</i> Returns the current horizontal position of the mouse in its parent's coordinate space.
j_getmousey	<i>int j_getmousey (int mouselistener);</i> Returns the current vertical position of the mouse in its parent's coordinate space.

Panel

j_panel	<i>int j_panel (int obj);</i> Creates a new panel component and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds panel obj to container cont
j_borderpanel	<i>int j_borderpanel (int obj , int type);</i> Creates a new borderpanel component with the style type and returns its event number.
j_button	<i>int j_button (int obj , char* label);</i> Creates a new button component with the specified label and returns its event number.
j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error -1 will be returned.
j_checkbox	<i>int j_checkbox (int obj , char* label);</i> Creates a new checkbox component with the specified label and returns its event number.
j_choice	<i>int j_choice (int obj);</i> Creates a new choice component and returns its event number.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to panel obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables panel obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the panel obj .
j_enable	<i>void j_enable (int obj);</i> enables the panel obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to panel obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of panel obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i>

	Returns the total pixel height of the actual font of panel obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of panel obj .
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Returns the width of the specified inset.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Returns the event number of the layoutmanager for containers obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of panel obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of panel obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of panel obj in its parent's coordinate space.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>void j_hide (int obj);</i> Hides the panel obj .
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to panel obj , and returns its event number.

j_label	<i>int j_label (int obj , char* label);</i> Creates a new label component with the specified label and returns its event number.
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>int j_list (int obj , int rows);</i> Creates a new list component with the specified number of rows and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to panel obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_pack	<i>void j_pack (int obj);</i> Resizes panel to the minimal size of contained components.
j_panel	<i>int j_panel (int obj);</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the panel .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Creates a new radiogroup and returns its event number.
j_release	<i>void j_release (int obj);</i> Releases panel obj from its parent component (container).
j_releaseall	<i>void j_releaseall (int obj);</i> Releases all components from panel obj .
j_scrollpane	<i>int j_scrollpane (int obj);</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>void j_setalign (int obj , int align);</i> Sets the alignment in panel obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Adds a borderlayout manager to panel obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves panel obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.

j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the panel 's obj cursor to the specified cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Adds a fixlayout manager to panel obj (default layout manager).
j_setflowfill	<i>void j_setflowfill (int obj , int bool);</i> Resizes all containing component to the height (width) of panel obj . Needs a flowlayout manager.
j_setflowlayout	<i>void j_setflowlayout (int obj , int align);</i> Adds a flowlayout manager to panel obj with the specified alignment .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to panel obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Adds a gridlayout manager to panel obj with the specified rows and columns .
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Set the insets to the specified values.
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Removes the current layout manager from panel obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the panel obj to the specified Position (xpos,ypos).

j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes panel obj to specified width and height .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Sets the vertical gap between components to hgap Pixel.
j_show	<i>void j_show (int obj);</i> Shows the panel obj .
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.

Popupmenu

j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_checkmenuitem	<i>int j_checkmenuitem (int obj , char* label);</i> creates a new checkmenuitem with the specified label and returns its event number.
j_disable	<i>void j_disable (int obj);</i> Disables popupmenu obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the popupmenu obj .
j_enable	<i>void j_enable (int obj);</i> enables the popupmenu obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of popupmenu 's label or text.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the popupmenu 's text or label.
j_menuitem	<i>int j_menuitem (int obj , char* label);</i> Creates a new menuitem with the specified label and returns its event number.
j_seperator	<i>void j_seperator (int obj);</i> Adds a separator bar to the popupmenu .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setshortcut	<i>void j_setshortcut (int obj , char chr);</i> Changes the shortcut chr of the popupmenu .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the popupmenu obj to str .

j_showpopup*void j_showpopup (int obj , int xpos , int ypos);*Shows the popupmenu at specified Position (**xpos,ypos**).

Printer

j_printer	<i>int j_printer (int frame);</i> Creates a new object, representing a paper of the printer.
j_cliprect	<i>void j_cliprect (int obj , int x , int y , int width , int height);</i> Changes current clipping region to the specified rectangle (x , y , width , height).
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the printer obj .
j_drawarc	<i>void j_drawarc (int obj , int x , int y , int rx , int ry , int arc1 , int arc2);</i> Draws an unfilled arc from angle arc1 to angle arc2 with the center (x , y) and the horizontal radius rx and the vertical radius ry .
j_drawcircle	<i>void j_drawcircle (int obj , int x , int y , int r);</i> Draws an unfilled circle with center (x , y) and radius x .
j_drawimage	<i>void j_drawimage (int obj , int image , int x , int y);</i> Copies the image, given by its eventnumber image , to position (x , y).
j_drawimagesource	<i>void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</i> Paints an image at Position (x , y) with width and height . The red, green and blue values of each pixel are given by the arrays r , g , b .
j_drawline	<i>void j_drawline (int obj , int x1 , int y1 , int x2 , int y2);</i> Draws a line connecting (x1 , y1) and (x2 , y2).
j_drawoval	<i>void j_drawoval (int obj , int x , int y , int rx , int ry);</i> Draws an unfilled oval with the center (x , y) and the horizontal radius rx and the vertical radius ry .
j_drawpixel	<i>void j_drawpixel (int obj , int x , int y);</i> Draws a pixel at (x , y).
j_drawpolygon	<i>void j_drawpolygon (int obj , int len , int* x , int* y);</i> Draws an unfilled polygon based on first len elements in x and y .
j_drawpolyline	<i>void j_drawpolyline (int obj , int len , int* x , int* y);</i> Draws a series of line segments based on first len elements in x and y .
j_drawrect	<i>void j_drawrect (int obj , int x , int y , int width , int height);</i> Draws an unfilled rectangle from (x , y) of size width x height .
j_drawroundrect	<i>void j_drawroundrect (int obj , int x , int y , int width , int height , int arcx , int arcy);</i>

Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

j_drawscaledimage *void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);*

Copy the contents of the rectangular area defined by **x**, **y**, width **sw**, and height **sh** of the **image** to position **(tx, ty)**. The area will be scaled to target width **th** and target height **th**.

j_drawstring *void j_drawstring (int obj , int x , int y , char* str);*

Draws text on screen at position **(x,y)**.

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

Draws an filled arc from angle **arc1** to angle **arc2** with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.

j_fillcircle *void j_fillcircle (int obj , int x , int y , int r);*

Draws an filled circle with center **(x, y)** and radius **x**.

j_filloval *void j_filloval (int obj , int x , int y , int rx , int ry);*

Draws an filled oval with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.

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

Draws an filled polygon based on first **len** elements in **x** and **y**.

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

Draws an filled rectangle from **(x,y)** of size **width** x **height**.

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

Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

j_print *void j_print (int obj);*

prints the printer .

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

Changes painting mode to XOR mode, if bool = J_TRUE . In this mode, drawing the same object in the same color at the same location twice has no net effect.

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

Moves the origin of drawing operations to **(x, y)**.

Radiobutton

j_radiobutton	<i>int j_radiobutton (int obj , char* label);</i> Creates a new radiobutton with the specified label and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds radiobutton obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to radiobutton obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables radiobutton obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the radiobutton obj .
j_enable	<i>void j_enable (int obj);</i> enables the radiobutton obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to radiobutton obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of radiobutton obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of radiobutton obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of radiobutton obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstate	<i>int j_getstate (int obj);</i> Returns J_TRUE , if radiobutton is selected, J_FALSE otherwise.
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the radiobutton 's text or label.

j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of radiobutton obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of radiobutton obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of radiobutton obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the radiobutton obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to radiobutton obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to radiobutton obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the radiobutton .
j_release	<i>void j_release (int obj);</i> Releases radiobutton obj from its parent component (container).
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves radiobutton obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the radiobutton 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to radiobutton obj .

j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the radiobutton obj to the specified Position (xpos,ypos).
j_setradiogroup	<i>int j_setradiogroup (int rbutton , , int rgroup);</i> Sets radiobuttons rbutton group to be the specified radiogroup rgroup . If the radiobuttons is already in a different radiogroup, it is first taken out of that group.
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes radiobutton obj to specified width and height .
j_setstate	<i>void j_setstate (int obj , int bool);</i> The radiobutton becomes selected, if bool is J_TRUE .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the radiobutton obj to str .
j_show	<i>void j_show (int obj);</i> Shows the radiobutton obj .

ScrollPane

j_scrollpane	<i>int j_scrollpane (int obj);</i> Creates a new scrollpane component and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds scrollpane obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to scrollpane obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables scrollpane obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the scrollpane obj .
j_enable	<i>void j_enable (int obj);</i> enables the scrollpane obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to scrollpane obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of scrollpane obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of scrollpane obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of scrollpane obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getviewportheight	<i>int j_getviewportheight (int obj);</i> Returns the height of the scrollpane 's obj port (the area that is shown)
j_getviewportwidth	<i>int j_getviewportwidth (int obj);</i> Returns the width of the scrollpane 's obj port (the area that is shown)

j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of scrollpane obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of scrollpane obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of scrollpane obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the scrollpane obj .
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to scrollpane obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to scrollpane obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the scrollpane .
j_release	<i>void j_release (int obj);</i> Releases scrollpane obj from its parent component (container).
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves scrollpane obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the scrollpane 's obj cursor to the specified cursor .

j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to scrollpane obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the scrollpane obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes scrollpane obj to specified width and height .
j_show	<i>void j_show (int obj);</i> Shows the scrollpane obj .
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.

Textarea

j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds textarea obj to container cont
j_appendtext	<i>void j_appendtext (int obj , char* text);</i> Appends the given text to the obj current text.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to textarea obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_delete	<i>void j_delete (int obj , int start , int end);</i> Deletes text from starting position start to ending position end .
j_disable	<i>void j_disable (int obj);</i> Disables textarea obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the textarea obj .
j_enable	<i>void j_enable (int obj);</i> enables the textarea obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to textarea obj , and returns its event number.
j_getcolumns	<i>void j_getcolumns (int obj);</i> Gets the number of columns in obj .
j_getcurpos	<i>int j_getcurpos (int obj);</i> Returns the position, in characters, of the text cursor.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of textarea obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of textarea obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of textarea obj .
j_getlength	<i>int j_getlength (int obj);</i>

	Returns the length of textarea 's label or text.
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getrows	<i>void j_getrows (int obj);</i> Gets the number of rows in obj .
j_getselend	<i>int j_getselend (int obj);</i> Returns the ending position of any selected text.
j_getselstart	<i>int j_getselstart (int obj);</i> Returns the initial position of any selected text.
j_getseltext	<i>char* j_getseltext (int obj , char* text);</i> Returns the currently selected text of textarea obj .
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the textarea 's text or label.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of textarea obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of textarea obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of textarea obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the textarea obj .
j_inserttext	<i>void j_inserttext (int obj , char* text , int pos);</i> Places additional text within the textarea at the given position pos .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to textarea obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i>

	<p>Adds a new mouse listener to textarea obj, and returns its event number. An event occurs, if the user action is of kind kind.</p>
j_popupmenu	<p><i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.</p>
j_print	<p><i>void j_print (int obj);</i> prints the textarea .</p>
j_release	<p><i>void j_release (int obj);</i> Releases textarea obj from its parent component (container).</p>
j_replacetext	<p><i>void j_replacetext (int obj , char* text , int start , int end);</i> Replaces the text from starting position start to ending position end with the given text.</p>
j_selectall	<p><i>void j_selectall (int obj);</i> Selects all the text in the textarea .</p>
j_selecttext	<p><i>void j_selecttext (int obj , int start , int end);</i> Selects text from starting position start to ending position end.</p>
j_setborderpos	<p><i>void j_setborderpos (int obj , int pos);</i> Moves textarea obj at a certain position. The outer container needs a border layout manager.</p>
j_setcolor	<p><i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r, g, b) values.</p>
j_setcolorbg	<p><i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r, g, b) values.</p>
j_setcolumns	<p><i>void j_setcolumns (int obj , int columns);</i> Sets the number of columns for obj to columns.</p>
j_setcurpos	<p><i>void j_setcurpos (int obj , int pos);</i> Change the location of the text cursor to the specified position pos.</p>
j_setcursor	<p><i>int j_setcursor (int obj , int cursor);</i> Changes the textarea 's obj cursor to the specified cursor.</p>
j_seteditable	<p><i>void j_seteditable (int obj , int bool);</i> Allows to make the textarea editable (bool=J_TRUE) or read-only (bool=J_FALSE).</p>
j_setfocus	<p><i>int j_setfocus (int obj);</i> Directs the input focus to textarea obj.</p>
j_setfont	<p><i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name, style and size.</p>

j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the textarea obj to the specified Position (xpos,ypos).
j_setrows	<i>void j_setrows (int obj , int rows);</i> Sets the number of rows for obj to rows .
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes textarea obj to specified width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the textarea obj to str .
j_show	<i>void j_show (int obj);</i> Shows the textarea obj .

Textfield

j_textfield	<i>int j_textfield (int obj , int columns);</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds textfield obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to textfield obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables textfield obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the textfield obj .
j_enable	<i>void j_enable (int obj);</i> enables the textfield obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to textfield obj , and returns its event number.
j_getcolumns	<i>void j_getcolumns (int obj);</i> Gets the number of columns in obj .
j_getcurpos	<i>int j_getcurpos (int obj);</i> Returns the position, in characters, of the text cursor.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of textfield obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of textfield obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of textfield obj .
j_getlength	<i>int j_getlength (int obj);</i> Returns the length of textfield 's label or text.
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.

j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getselend	<i>int j_getselend (int obj);</i> Returns the ending position of any selected text.
j_getselstart	<i>int j_getselstart (int obj);</i> Returns the initial position of any selected text.
j_getseltext	<i>char* j_getseltext (int obj , char* text);</i> Returns the currently selected text of textfield obj .
j_gettext	<i>char* j_gettext (int obj , char* str);</i> returns the textfield 's text or label.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of textfield obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of textfield obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of textfield obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the textfield obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to textfield obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to textfield obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the textfield .
j_release	<i>void j_release (int obj);</i> Releases textfield obj from its parent component (container).

j_selectall	<i>void j_selectall (int obj);</i> Selects all the text in the textfield .
j_selecttext	<i>void j_selecttext (int obj , int start , int end);</i> Selects text from starting position start to ending position end .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves textfield obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcolumns	<i>void j_setcolumns (int obj , int columns);</i> Sets the number of columns for obj to columns .
j_setcurpos	<i>void j_setcurpos (int obj , int pos);</i> Change the location of the text cursor to the specified position pos .
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the textfield 's obj cursor to the specified cursor .
j_setechochar	<i>void j_setechochar (int obj , char chr);</i> Changes the character chr that is used to echo all user input in the textfield .
j_seteditable	<i>void j_seteditable (int obj , int bool);</i> Allows to make the textfield editable (bool =J_TRUE) or read-only (bool =J_FALSE).
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to textfield obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .

j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the textfield obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes textfield obj to specified width and height .
j_settext	<i>void j_settext (int obj , char* str);</i> Sets the content or the label of the textfield obj to str .
j_show	<i>void j_show (int obj);</i> Shows the textfield obj .

Vscrollbar

j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds vscrollbar obj to container cont
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to vscrollbar obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables vscrollbar obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the vscrollbar obj .
j_enable	<i>void j_enable (int obj);</i> enables the vscrollbar obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to vscrollbar obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of vscrollbar obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i> Returns the total pixel height of the actual font of vscrollbar obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of vscrollbar obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getvalue	<i>int j_getvalue (int obj);</i> Returns the current setting of the scrollbar.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of vscrollbar obj .

j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of scrollbar obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of scrollbar obj in its parent's coordinate space.
j_hide	<i>void j_hide (int obj);</i> Hides the scrollbar obj .
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to scrollbar obj , and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to scrollbar obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the scrollbar .
j_release	<i>void j_release (int obj);</i> Releases scrollbar obj from its parent component (container).
j_setblockinc	<i>int j_setblockinc (int obj , int val);</i> Changes the block increment amount for the scrollbar to val .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves scrollbar obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.
j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the scrollbar 's obj cursor to the specified cursor .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to scrollbar obj .

j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setmax	<i>int j_setmax (int obj , int val);</i> Changes the maximum value for the scrollbar to val .
j_setmin	<i>int j_setmin (int obj , int val);</i> Changes the minimum value for the scrollbar to val .
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the scrollbar obj to the specified Position (xpos,ypos).
j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes scrollbar obj to specified width and height .
j_setslidesize	<i>int j_setslidesize (int obj , int val);</i> Changes the slide size to val .
j_setunitinc	<i>int j_setunitinc (int obj , int val);</i> Changes the unit increment amount for the scrollbar to val .
j_setvalue	<i>void j_setvalue (int obj , int val);</i> Changes the current value of the scrollbar to val .
j_show	<i>void j_show (int obj);</i> Shows the scrollbar obj .

Window

j_window	<i>int j_window (int obj);</i> Creates a new simple window and returns its event number.
j_add	<i>void j_add (int obj , int cont);</i> Adds window obj to container cont
j_borderpanel	<i>int j_borderpanel (int obj , int type);</i> Creates a new borderpanel component with the style type and returns its event number.
j_button	<i>int j_button (int obj , char* label);</i> Creates a new button component with the specified label and returns its event number.
j_canvas	<i>int j_canvas (int obj , int width , int height);</i> Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error -1 will be returned.
j_checkbox	<i>int j_checkbox (int obj , char* label);</i> Creates a new checkbox component with the specified label and returns its event number.
j_choice	<i>int j_choice (int obj);</i> Creates a new choice component and returns its event number.
j_componentlistener	<i>int j_componentlistener (int obj , int kind);</i> Adds a new componentlistener to window obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_disable	<i>void j_disable (int obj);</i> Disables window obj so that it is unresponsive to user interactions
j_dispose	<i>void j_dispose (int obj);</i> Releases the resources of the window obj .
j_enable	<i>void j_enable (int obj);</i> enables the window obj .
j_focuslistener	<i>int j_focuslistener (int obj);</i> Adds a new focus listener to window obj , and returns its event number.
j_getfontascent	<i>int j_getfontascent (int obj);</i> Returns the ascent (space above the baseline) of the actual font of window obj .
j_getfontheight	<i>int j_getfontheight (int obj);</i>

	Returns the total pixel height of the actual font of window obj .
j_getheight	<i>int j_getheight (int obj);</i> Returns the height of window obj .
j_getinsets	<i>int j_getinsets (int obj , int side);</i> Returns the width of the specified inset.
j_getlayoutid	<i>int j_getlayoutid (int obj);</i> Returns the event number of the layoutmanager for containers obj .
j_getparent	<i>int j_getparent (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparentid	<i>int j_getparentid (int obj);</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getwidth	<i>int j_getwidth (int obj);</i> Returns the width of window obj .
j_getxpos	<i>int j_getxpos (int obj);</i> Returns the current horizontal position of window obj in its parent's coordinate space.
j_getypos	<i>int j_getypos (int obj);</i> Returns the current vertical position of window obj in its parent's coordinate space.
j_graphicbutton	<i>int j_graphicbutton (int obj , char* filename);</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>int j_graphiclabel (int obj , char* str);</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>void j_hide (int obj);</i> Hides the window obj .
j_hscrollbar	<i>int j_hscrollbar (int obj);</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>int j_isparent (int obj , int cont);</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>int j_isvisible (int obj);</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>int j_keylistener (int obj);</i> Adds a new key listener to window obj , and returns its event number.

j_label	<i>int j_label (int obj , char* label);</i> Creates a new label component with the specified label and returns its event number.
j_line	<i>int j_line (int obj , int orient , int style , int length);</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>int j_list (int obj , int rows);</i> Creates a new list component with the specified number of rows and returns its event number.
j_mouselistener	<i>int j_mouselistener (int obj , int kind);</i> Adds a new mouse listener to window obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_pack	<i>void j_pack (int obj);</i> Resizes window to the minimal size of contained components.
j_panel	<i>int j_panel (int obj);</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>int j_popupmenu (int obj , char* label);</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>void j_print (int obj);</i> prints the window .
j_radiogroup	<i>int j_radiogroup (int obj);</i> Creates a new radiogroup and returns its event number.
j_release	<i>void j_release (int obj);</i> Releases window obj from its parent component (container).
j_releaseall	<i>void j_releaseall (int obj);</i> Releases all components from window obj .
j_scrollpane	<i>int j_scrollpane (int obj);</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>void j_setalign (int obj , int align);</i> Sets the alignment in window obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>void j_setborderlayout (int obj);</i> Adds a borderlayout manager to window obj .
j_setborderpos	<i>void j_setborderpos (int obj , int pos);</i> Moves window obj at a certain position. The outer container needs a border layout manager.
j_setcolor	<i>void j_setcolor (int obj , int r , int g , int b);</i> Sets the foreground color to the (r , g , b) values.

j_setcolorbg	<i>void j_setcolorbg (int obj , int r , int g , int b);</i> Sets the background color to the (r , g , b) values.
j_setcursor	<i>int j_setcursor (int obj , int cursor);</i> Changes the window 's obj cursor to the specified cursor .
j_setfixlayout	<i>void j_setfixlayout (int obj);</i> Adds a fixlayout manager to window obj (default layout manager).
j_setflowfill	<i>void j_setflowfill (int obj , int bool);</i> Resizes all containing component to the height (width) of window obj . Needs a flowlayout manager.
j_setflowlayout	<i>void j_setflowlayout (int obj , int align);</i> Adds a flowlayout manager to window obj with the specified alignment .
j_setfocus	<i>int j_setfocus (int obj);</i> Directs the input focus to window obj .
j_setfont	<i>void j_setfont (int obj , int name , int style , int size);</i> Changes the font to the given characteristics name , style and size .
j_setfontname	<i>void j_setfontname (int obj , int name);</i> Changes the font to the given name .
j_setfontsize	<i>void j_setfontsize (int obj , int size);</i> Changes the font to the given size .
j_setfontstyle	<i>void j_setfontstyle (int obj , int style);</i> Changes the font to the given style .
j_setgridlayout	<i>void j_setgridlayout (int obj , int row , int col);</i> Adds a gridlayout manager to window obj with the specified rows and columns .
j_sethgap	<i>void j_sethgap (int obj , int hgap);</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>void j_setinsets (int obj , int top , int bottom , int left , int right);</i> Set the insets to the specified values.
j_setnamedcolor	<i>void j_setnamedcolor (int obj , int color);</i> Sets the foreground color to a predefined color .
j_setnamedcolorbg	<i>void j_setnamedcolorbg (int obj , int color);</i> Sets the background color to a predefined color .
j_setnolayout	<i>void j_setnolayout (int obj);</i> Removes the current layout manager from window obj .
j_setpos	<i>void j_setpos (int obj , int xpos , int ypos);</i> Relocates the window obj to the specified Position (xpos , ypos).

j_setsize	<i>void j_setsize (int obj , int width , int height);</i> Resizes window obj to specified width and height .
j_setvgap	<i>void j_setvgap (int obj , int vgap);</i> Sets the vertical gap between components to hgap Pixel.
j_show	<i>void j_show (int obj);</i> Shows the window obj .
j_textarea	<i>int j_textarea (int obj , int rows , int columns);</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>int j_textfield (int obj , int columns);</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>int j_vscrollbar (int obj);</i> Creates a new vertical scrollbar and returns its event number.
j_windowlistener	<i>int j_windowlistener (int window , int kind);</i> Adds a new windowlistener to obj , and returns its event number. An event occurs, if the user action is of kind kind .

Kapitel 2

Functions

add

Synopsis	void j_add (int obj , int cont);
----------	---

Arguments	obj int cont int
-----------	--

Description	Adds component obj to container cont
-------------	--

Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield
---------	---

additem

Synopsis `void j_additem (int obj , char* str);`

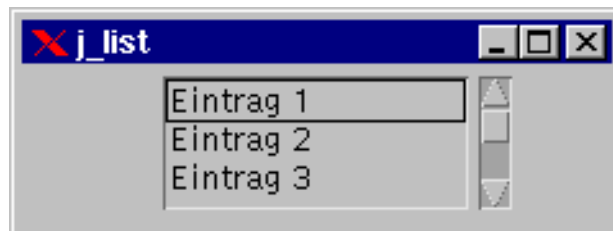
Arguments `obj` `int`
 `str` `char*`

Description adds a new item containing **str** to component **obj**.

Targets List, Choice

Example

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



alertbox

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

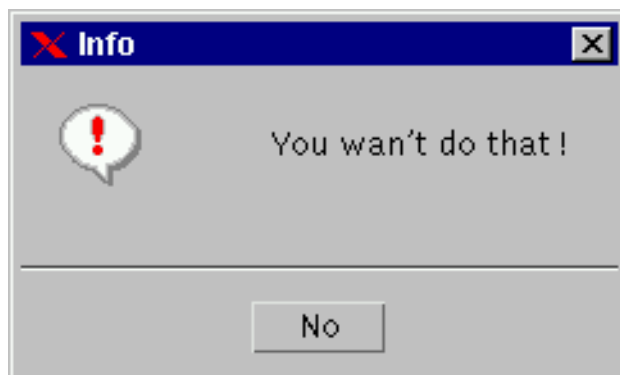
Arguments obj int
 title char*
 text char*
 button char*

Description Shows a alertbox with the specified **title**, **text** and **button**. Alert-boxes are modal dialogs, the application is blocked until the button or the closeicon is clicked. The return value is 0 if the closeicon is clicked and 1 if the buttons is used.

Targets Frame

Example

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



appendtext

Synopsis	void j_appendtext (int obj , char* text);	
Arguments	obj	int
	text	char*
Description	Appends the given text to the obj current text.	
Targets	Textarea	

beep

Synopsis	<code>void j_beep ();</code>
Description	Emits an audio beep.

borderpanel

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

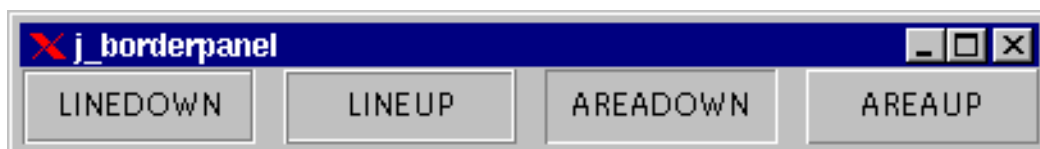
Arguments `obj` `int`
 `type` `int`

Description Creates a new borderpanel component with the style **type** and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

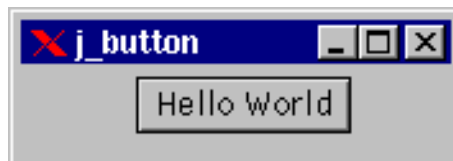
Example

```
:  
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);  
:
```



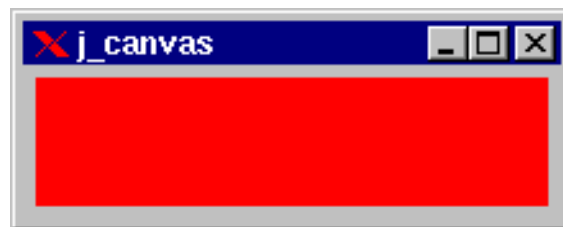
button

Synopsis	<code>int j_button (int obj , char* label);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>label</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>label</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>label</code>	<code>char*</code>				
Description	Creates a new button component with the specified label and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: frame = j_frame("j_button"); button = j_button(frame,"Hello World"); :</pre>				



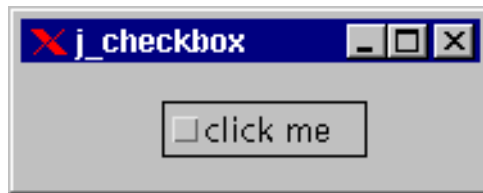
canvas

Synopsis	<code>int j_canvas (int obj , int width , int height);</code>						
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>width</code></td><td><code>int</code></td></tr><tr><td><code>height</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>width</code>	<code>int</code>	<code>height</code>	<code>int</code>
<code>obj</code>	<code>int</code>						
<code>width</code>	<code>int</code>						
<code>height</code>	<code>int</code>						
Description	Creates a new canvas component with the given width and height and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error <code>-1</code> will be returned.						
Targets	Panel, Borderpanel, Window, Dialog, Frame						
Example	<pre>: canvas = j_canvas(frame,200,50); j_setnamedcolorbg(canvas,J_RED); :</pre>						



checkbox

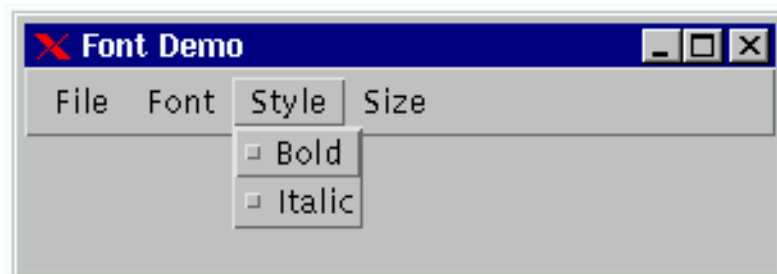
Synopsis	<code>int j_checkbox (int obj , char* label);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>label</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>label</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>label</code>	<code>char*</code>				
Description	Creates a new checkbox component with the specified label and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: frame = j_frame("j_checkbox"); checkbox = j_checkbox(frame,"click me"); :</pre>				



checkmenuitem

Synopsis	<code>int j_checkmenuitem (int obj , char* label);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>label</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>label</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>label</code>	<code>char*</code>				
Description	creates a new checkmenuitem with the specified label and returns its event number.				
Targets	Menu, Popupmenu, Helpmenu				
Example					

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



choice

Synopsis `int j_choice (int obj);`

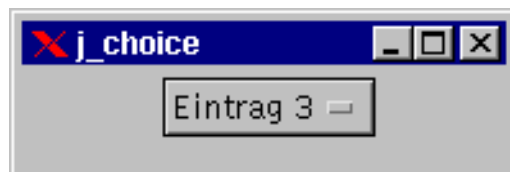
Arguments `obj` `int`

Description Creates a new choice component and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



choicebox2

Synopsis

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

Arguments

obj	int
title	char*
text	char*
button1	char*
button2	char*

Description

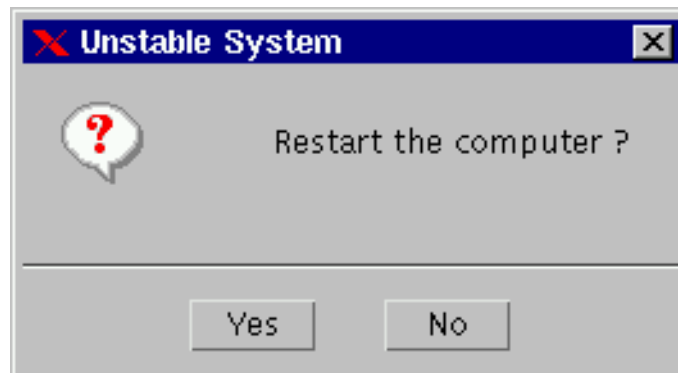
Shows a choicebox with the specified **title**, **text** and two buttons. Choiceboxes are modal dialogs, the application is blocked until a button or the closeicon is clicked. The focus is set to the first button. The return value is 0 if the closeicon is clicked, 1 for the first button and 2 for the second one.

Targets

Frame

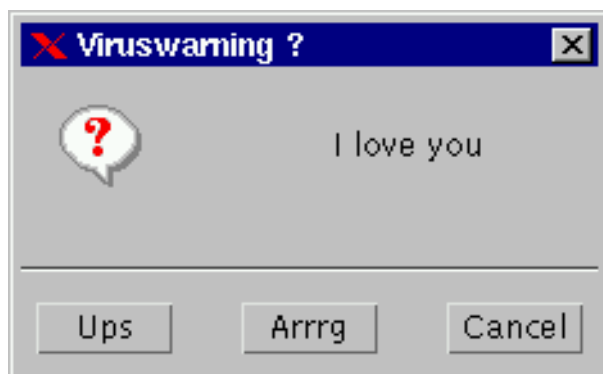
Example

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



checkboxbox3

Synopsis	void j_checkbox3 (int obj , char* title , char* text , char* button1 , char* button2 , char* button3);												
Arguments	<table> <tr><td>obj</td><td>int</td></tr> <tr><td>title</td><td>char*</td></tr> <tr><td>text</td><td>char*</td></tr> <tr><td>button1</td><td>char*</td></tr> <tr><td>button2</td><td>char*</td></tr> <tr><td>button3</td><td>char*</td></tr> </table>	obj	int	title	char*	text	char*	button1	char*	button2	char*	button3	char*
obj	int												
title	char*												
text	char*												
button1	char*												
button2	char*												
button3	char*												
Description	Shows a checkbox with the specified title , text and three buttons. Choiceboxes are modal dialogs, the application is blocked until a button or the closeicon is clicked. The focus is set to the first button. The return value is 0 if the closeicon is clicked, 1 for the first button, 2 for the second and 3 for the third one.												
Targets	Frame												
Example	<pre> : retval = j_checkbox2(frame,"Viruswarning ?","I love you", "Ups","Arrrg","Cancel"); : </pre>												



cliprect

Synopsis	<code>void j_cliprect (int obj , int x , int y , int width , int height);</code>										
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>x</code></td><td><code>int</code></td></tr><tr><td><code>y</code></td><td><code>int</code></td></tr><tr><td><code>width</code></td><td><code>int</code></td></tr><tr><td><code>height</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>x</code>	<code>int</code>	<code>y</code>	<code>int</code>	<code>width</code>	<code>int</code>	<code>height</code>	<code>int</code>
<code>obj</code>	<code>int</code>										
<code>x</code>	<code>int</code>										
<code>y</code>	<code>int</code>										
<code>width</code>	<code>int</code>										
<code>height</code>	<code>int</code>										
Description	Changes current clipping region to the specified rectangle (x , y , width , height).										
Targets	Canvas, Image, Printer										

componentlistener

Synopsis	int j_componentlistener (int obj , int kind);
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>kind</div> <div>int</div> </div>
Description	<p>Adds a new componentlistener to component obj, and returns its event number. An event occurs, if the user action is of kind kind. Possible values for kind:</p> <ul style="list-style-type: none"> • J_RESIZED : An event occurs when the component has been resized. • J_HIDDEN : An event occurs when the component has been hidden. • J_SHOWN : An event occurs when the component has been shown.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

connect

Synopsis `int j_connect (char* hostname);`

Arguments `hostname char*`

Description Connects a running japi kernel on host **hostname**.

Example

```
:
if( ! j_connect("atan.japi.de"))

or

if( ! j_connect("127.0.0.1"))
:
```

delete

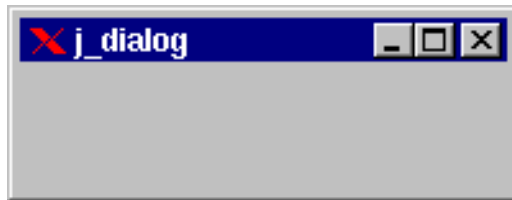
Synopsis	void j_delete (int obj , int start , int end);	
Arguments	obj	int
	start	int
	end	int
Description	Deletes text from starting position start to ending position end .	
Targets	Textarea	

deselect

Synopsis	int j_deselect (int obj , int item);	
Arguments	obj	int
	item	int
Description	Deselects the item at the designated position item , if selected.	
Targets	List	

dialog

Synopsis	<code>int j_dialog (int obj , char* label);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>label</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>label</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>label</code>	<code>char*</code>				
Description	Creates a new dialog window with the specified label and returns its event number.				
Targets	Frame				
Example	<pre>: dialog = j_dialog(frame,"j_dialog"); j_setsize(dialog,200,80); j_show(dialog); :</pre>				



disable

Synopsis	<code>void j_disable (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Disables component obj so that it is unresponsive to user interactions
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, MenuItem, Check-BoxMenuItem, Menu, HelpMenu, Popupmenu

dispose

Synopsis	<code>void j_dispose (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Releases the resources of the component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Canvas, Image, Printer, Keylistener, Focuslistener, Moueslistener

drawarc

Synopsis

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

Arguments

obj	int
x	int
y	int
rx	int
ry	int
arc1	int
arc2	int

Description

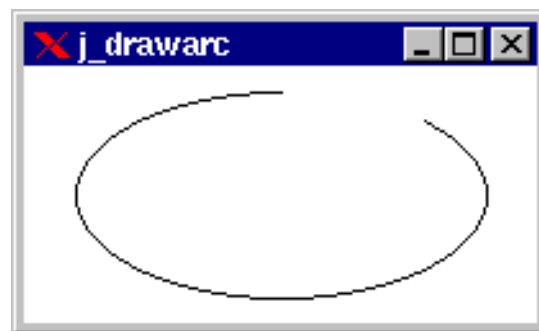
Draws an unfilled arc from angle **arc1** to angle **arc2** with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets

Canvas, Image, Printer

Example

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



drawcircle

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

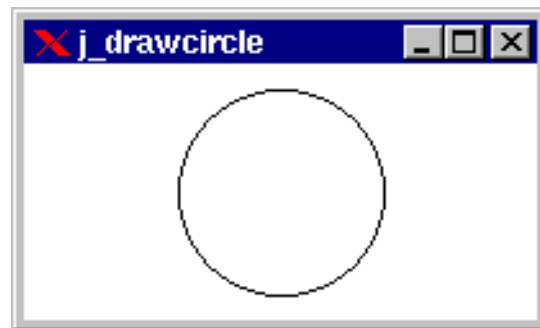
Arguments obj int
 x int
 y int
 r int

Description Draws an unfilled circle with center (**x**, **y**) and radius **x**.

Targets Canvas, Image, Printer

Example

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



drawimage

Synopsis	void j_drawimage (int obj , int image , int x , int y);	
Arguments	obj	int
	image	int
	x	int
	y	int
Description	Copies the image, given by its eventnumber image , to position (x , y).	
Targets	Canvas, Image, Printer	

drawimagesource

Synopsis	void j_drawimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);	
Arguments	obj	int
	x	int
	y	int
	w	int
	h	int
	r	int*
	g	int*
	b	int*
Description	Paints an image at Position (x , y ,) with width and height . The red, green and blue values of each pixel are given by the arrays r , g , b .	
Targets	Canvas, Image, Printer	

drawline

Synopsis

```
void j_drawline ( int obj , int x1 , int y1 , int x2 , int y2 );
```

Arguments

obj	int
x1	int
y1	int
x2	int
y2	int

Description

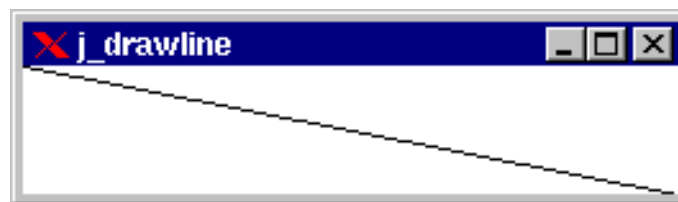
Draws a line connecting **(x1,y1)** and **(x2,y2)**.

Targets

Canvas, Image, Printer

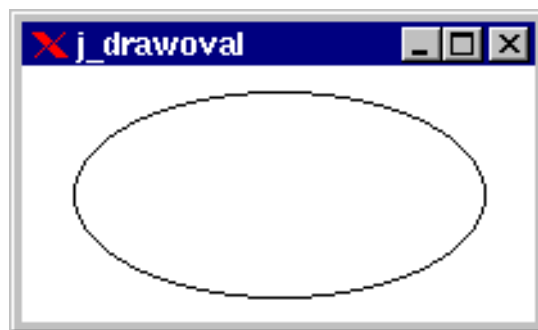
Example

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



drawoval

Synopsis	<code>void j_drawoval (int obj , int x , int y , int rx , int ry);</code>										
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>x</code></td><td><code>int</code></td></tr><tr><td><code>y</code></td><td><code>int</code></td></tr><tr><td><code>rx</code></td><td><code>int</code></td></tr><tr><td><code>ry</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>x</code>	<code>int</code>	<code>y</code>	<code>int</code>	<code>rx</code>	<code>int</code>	<code>ry</code>	<code>int</code>
<code>obj</code>	<code>int</code>										
<code>x</code>	<code>int</code>										
<code>y</code>	<code>int</code>										
<code>rx</code>	<code>int</code>										
<code>ry</code>	<code>int</code>										
Description	Draws an unfilled oval with the center (x , y) and the horizontal radius rx and the vertical radius ry .										
Targets	Canvas, Image, Printer										
Example	<pre>: canvas = j_canvas(frame,200,100); j_drawoval(canvas,100,50,80,40); :</pre>										



drawpixel

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

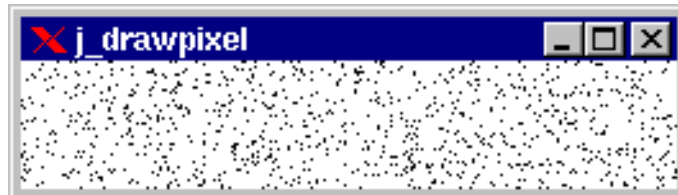
Arguments `obj` `int`
 `x` `int`
 `y` `int`

Description Draws a pixel at (**x,y**).

Targets Canvas, Image, Printer

Example

```
:  
canvas = j_canvas(frame,256,50);  
for(i=0;i<1000;i++)  
    j_drawpixel(canvas,j_random()%256,,j_random()%256);  
:
```



drawpolygon

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

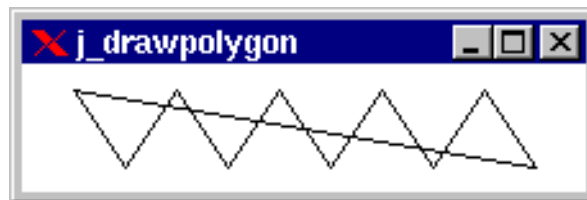
Arguments obj int
 len int
 x int*
 y int*

Description Draws an unfilled polygon based on first **len** elements in **x** and **y**.

Targets Canvas, Image, Printer

Example

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



drawpolyline

Synopsis

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

Arguments

obj	int
len	int
x	int*
y	int*

Description

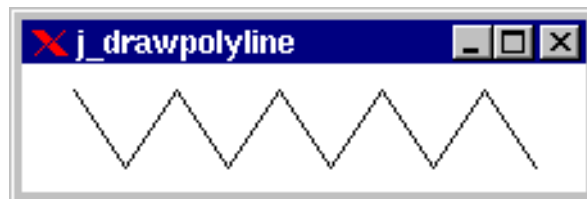
Draws a series of line segments based on first **len** elements in **x** and **y**.

Targets

Canvas, Image, Printer

Example

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



drawrect

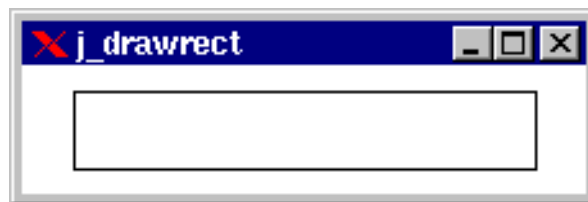
Synopsis `void j_drawrect (int obj , int x , int y , int width , int height);`

Arguments obj int
 x int
 y int
 width int
 height int

Description Draws an unfilled rectangle from (**x,y**) of size **width** x **height**.

Targets Canvas, Image, Printer

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



drawroundrect

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

Arguments

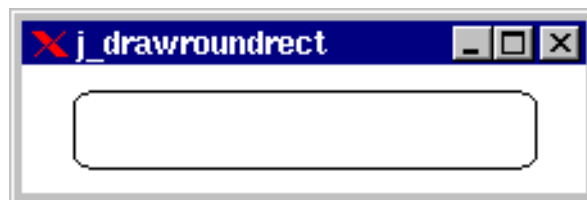
<code>obj</code>	<code>int</code>
<code>x</code>	<code>int</code>
<code>y</code>	<code>int</code>
<code>width</code>	<code>int</code>
<code>height</code>	<code>int</code>
<code>arcx</code>	<code>int</code>
<code>arcy</code>	<code>int</code>

Description Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

Targets Canvas, Image, Printer

Example

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



drawscaledimage

Synopsis	void j_drawscaledimage (int obj , int image , int sx , int sy , int sw , int sh , int tx , int ty , int tw , int th);	
Arguments	obj	int
	image	int
	sx	int
	sy	int
	sw	int
	sh	int
	tx	int
	ty	int
	tw	int
	th	int
Description	Copy the contents of the rectangular area defined by x, y, width sw , and height sh of the image to position (tx, ty . The area will be scaled to target width th and target height th .	
Targets	Canvas, Image, Printer	

drawstring

Synopsis `void j_drawstring (int obj , int x , int y , char* str);`

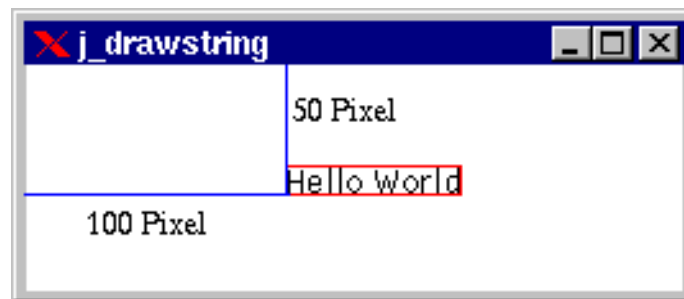
Arguments obj int
 x int
 y int
 str char*

Description Draws text on screen at position (x,y).

Targets Canvas, Image, Printer

Example

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

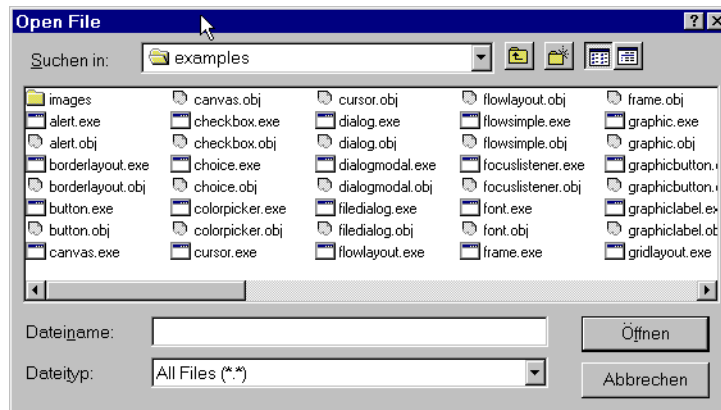


enable

Synopsis	<code>void j_enable (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	enables the component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Menuitem, Check-BoxMenuitem, Menu, HelpMenu, Popupmenu

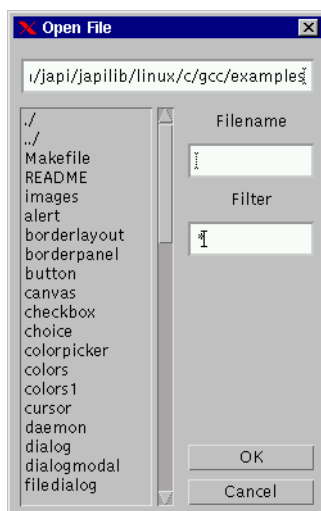
filedialog

Synopsis	<pre>char* j_filedialog (int frame , char* title , char* directory , char* filename);</pre>								
Arguments	<table> <tr><td>frame</td><td>int</td></tr> <tr><td>title</td><td>char*</td></tr> <tr><td>directory</td><td>char*</td></tr> <tr><td>filename</td><td>char*</td></tr> </table>	frame	int	title	char*	directory	char*	filename	char*
frame	int								
title	char*								
directory	char*								
filename	char*								
Description	Opens a filedialog box in the specified directory with the specified title and returns the selected filename .								
Targets	Frame								
Example	<pre>: filename = j_filedialog(frame,"Open File","..",filename); :</pre>								



fileselector

Synopsis	char* j_fileselector (int frame , char* title , char* filter , char* filename);								
Arguments	<table> <tr><td>frame</td><td>int</td></tr> <tr><td>title</td><td>char*</td></tr> <tr><td>filter</td><td>char*</td></tr> <tr><td>filename</td><td>char*</td></tr> </table>	frame	int	title	char*	filter	char*	filename	char*
frame	int								
title	char*								
filter	char*								
filename	char*								
Description	Opens a fileselector box with the preselected filename and the specified title and returns the selected filename . filter specifies the Filename Filter. A Fileselector can be used with output redirections via <code>j_connect()</code> ;								
Targets	Frame								
Example	<pre> : filename = j_fileselect(frame,"Open File","*",filename); : </pre>								



fillarc

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

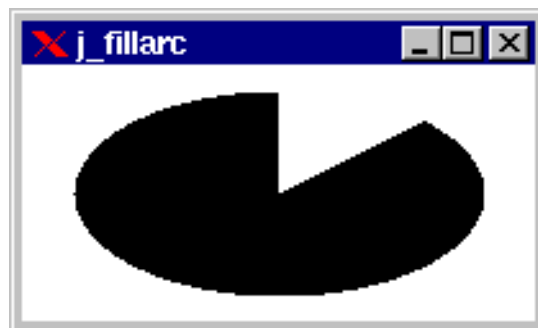
Arguments obj int
 x int
 y int
 rx int
 ry int
 arc1 int
 arc2 int

Description Draws an filled arc from angle **arc1** to angle **arc2** with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets Canvas, Image, Printer

Example

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



fillcircle

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

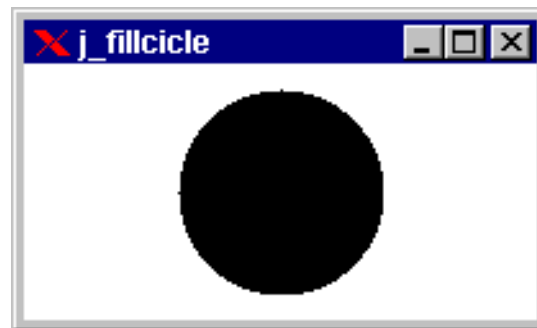
Arguments `obj` `int`
 `x` `int`
 `y` `int`
 `r` `int`

Description Draws an filled circle with center (**x**, **y**) and radius **x**.

Targets Canvas, Image, Printer

Example

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



filloval

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

Arguments

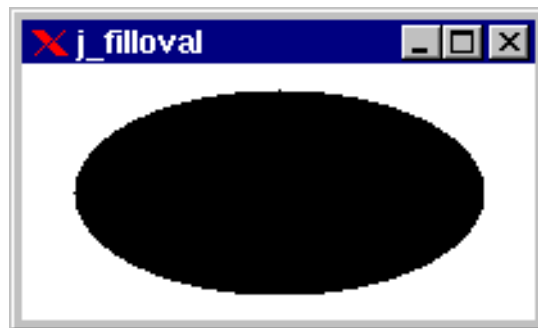
<code>obj</code>	<code>int</code>
<code>x</code>	<code>int</code>
<code>y</code>	<code>int</code>
<code>rx</code>	<code>int</code>
<code>ry</code>	<code>int</code>

Description Draws an filled oval with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets Canvas, Image, Printer

Example

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



fillpolygon

Synopsis	<code>void j_fillpolygon (int obj , int len , int* x , int* y);</code>
Arguments	<div> <div>obj</div> <div>len</div> <div>x</div> <div>y</div> </div> <div> <div>int</div> <div>int</div> <div>int*</div> <div>int*</div> </div>
Description	Draws an filled polygon based on first len elements in x and y .
Targets	Canvas, Image, Printer

Example

```

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

```



fillrect

Synopsis

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

Arguments

obj	int
x	int
y	int
width	int
height	int

Description

Draws an filled rectangle from **(x,y)** of size **width** x **height**.

Targets

Canvas, Image, Printer

Example

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



fillroundrect

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

Arguments

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

Description Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

Targets Canvas, Image, Printer

Example

```

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

```

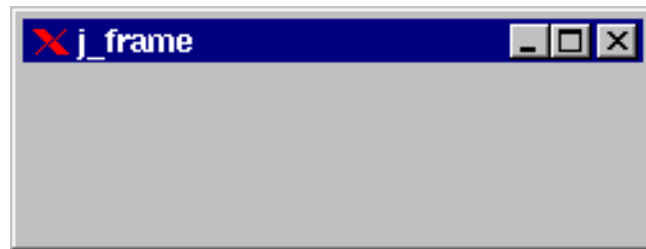


focuslistener

Synopsis	int j_focuslistener (int obj);
Arguments	obj int
Description	Adds a new focus listener to component obj , and returns its event number.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

frame

Synopsis	<code>int j_frame (char* label);</code>
Arguments	<code>label</code> <code>char*</code>
Description	Creates a new frame component with the specified label and returns its event number.
Example	<pre>: frame = j_frame("j_frame"); j_show(frame); :</pre>



getaction

Synopsis

```
int j_getaction ( );
```

Description

returns the next event, or 0 if no event available

getcolumns

Synopsis void **j_getcolumns** (int obj);

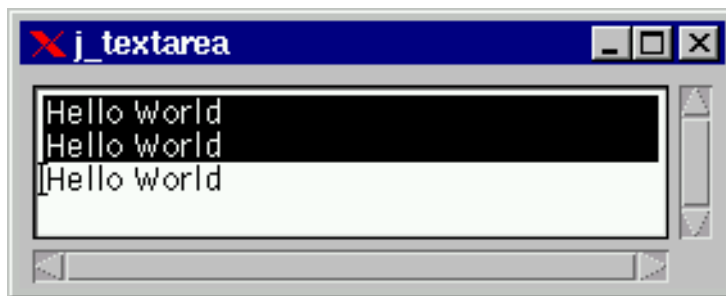
Arguments obj int

Description Gets the number of columns in **obj**.

Targets Textarea, Textfield, Gridlayout

Example

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



getcurpos

Synopsis	<code>int j_getcurpos (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the position, in characters, of the text cursor.
Targets	Textarea, Textfield

getfontascent

Synopsis	int j_getfontascent (int obj);
Arguments	obj int
Description	Returns the ascent (space above the baseline) of the actual font of component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

getfontheight

Synopsis	int j_getfontheight (int obj);
Arguments	obj int
Description	Returns the total pixel height of the actual font of component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

getheight

Synopsis `int j_getheight (int obj);`

Arguments `obj` `int`

Description Returns the height of component **obj**.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Image

Example

```
:  
label = j_getlabel(frame,"Hello World");  
printf("%s",j_getheight(label));  
:  
  
> 22
```

getimage

Synopsis	int j_getimage (int obj);
Arguments	obj int
Description	Copy the contents of component obj into an image and return its eventnumber.
Targets	Canvas, Image

getimagesource

Synopsis	<pre>int j_getimagesource (int obj , int x , int y , int w , int h , int* r , int* g , int* b);</pre>	
Arguments	<div>obj</div> <div>x</div> <div>y</div> <div>w</div> <div>h</div> <div>r</div> <div>g</div> <div>b</div>	<div>int</div> <div>int</div> <div>int</div> <div>int</div> <div>int</div> <div>int*</div> <div>int*</div> <div>int*</div>
Description	<p>Returns an image of the specified size (x, y, width, height) of component . The red, green and blue values of each pixel will be stored in r, g, b</p>	
Targets	Canvas, Image	

getinsets

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

Arguments `obj` `int`
 `side` `int`

Description Returns the width of the specified inset. **side** can take the following values:

- J_TOP: returns the height of the top inset.
- J_BOTTOM: returns the height of the bottom inset.
- J_LEFT: returns the width of the left inset.
- J_RIGHT: returns the width of the right inset.

Targets Panel, Borderpanel, Window, Dialog, Frame

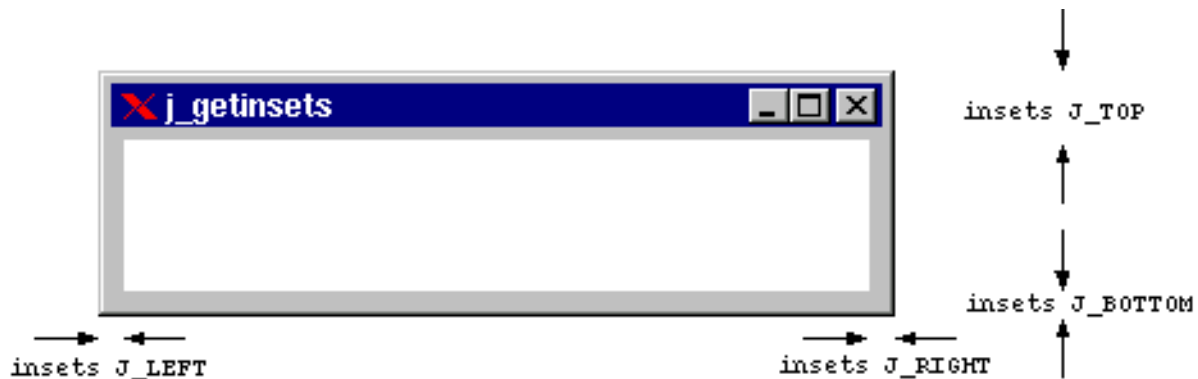
Example

```

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

> 25 5 5 6

```



getitem

Synopsis `char* j_getitem (int obj , int item , char* str);`

Arguments `obj int`
 `item int`
 `str char*`

Description returns the label of the given **item**.

Targets List, Choice

getitemcount

Synopsis	<code>int j_getitemcount (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the number of items of component obj .
Targets	List, Choice

getkeychar

Synopsis	int j_getkeychar (int obj);
Arguments	obj int
Description	Returns the ascii value of the last pressed key.
Targets	KeyListener

getkeycode

Synopsis	<code>int j_getkeycode (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the integer key code of the last pressed key.
Targets	KeyListener

getlayoutid

Synopsis	<code>int j_getlayoutid (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the event number of the layoutmanager for containers obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: j_setgridlayout(frame,2,2); grid = j_getlayoutid(frame); :</pre>

getlength

Synopsis	<code>int j_getlength (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the length of component 's label or text.
Targets	Textarea, Textfield, Dialog, Frame, Button, MenuItem, CheckBox- MenuItem, Menu, HelpMenu, Popupmenu

getmousebutton

Synopsis	int j_getmousebutton (int mouset listener);
Arguments	mouset listenerint
Description	Returns the latest used mousebutton. The return value is: <ul style="list-style-type: none">• J_LEFT left mousebutton• J_CENTER middle mousebutton• J_RIGHT right mousebutton
Targets	Mouset listener

getmousex

Synopsis	int j_getmousex (int mouset listener);
Arguments	mouset listenerint
Description	Returns the current horizontal position of the mouse in its parent's coordinate space.
Targets	Mouset listener

getmousey

Synopsis	int j_getmousey (int mouset listener);
Arguments	mouset listenerint
Description	Returns the current vertical position of the mouse in its parent's coordinate space.
Targets	Mouset listener

getparent

Synopsis `int j_getparent (int obj);`

Arguments `obj` `int`

Description Returns the parent event number of component **obj**. If **obj** is a frame `-1` will be returned.

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

Example

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



getparentid

Synopsis	<code>int j_getparentid (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the parent event number of component obj . If obj is a frame <code>-1</code> will be returned.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Example

```

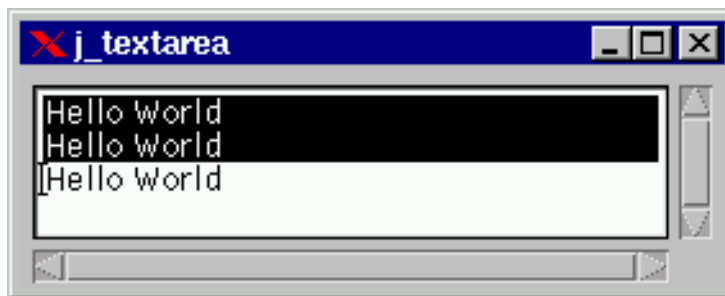
:
radio1      = j_radiobutton(j_radiogroup(frame),"Radiobutton 1");
radio2      = j_radiobutton(j_getparentid(radio1),"Radiobutton 2");
:

```



getrows

Synopsis	<code>void j_getrows (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Gets the number of rows in obj .
Targets	Textarea, Gridlayout
Example	<pre>: text = j_text(frame,30,4); j_getrows(text); : > 4</pre>



getscaledimage

Synopsis	int j_getscaledimage (int obj , int x , int y , int sw , int sh , int tw , int th);	
Arguments	obj	int
	x	int
	y	int
	sw	int
	sh	int
	tw	int
	th	int
Description	Copy the contents of the rectangular area defined by x , y , width sw , and height sh into an image and return its eventnumber. The image will be scaled to target width th and target height th .	
Targets	Canvas, Image	

getscreenheight

Synopsis

```
int j_getscreenheight ( );
```

Description

Returns the screens height in pixel. If a virtual screen is installed, the virtual height will be returned.

Example

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

getscreenwidth

Synopsis

```
int j_getscreenwidth ( );
```

Description

Returns the screens width in pixel. If a virtual screen is installed, the virtual width will be returned.

Example

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

getselect

Synopsis	int j_getselect (int obj);
Arguments	obj int
Description	Returns the position of currently selected item.
Targets	List, Choice

getselend

Synopsis	<code>int j_getselend (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the ending position of any selected text.
Targets	Textarea, Textfield

getselstart

Synopsis	<code>int j_getselstart (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the initial position of any selected text.
Targets	Textarea, Textfield

getselttext

Synopsis	<code>char* j_getselttext (int obj , char* text);</code>	
Arguments	<code>obj</code>	<code>int</code>
	<code>text</code>	<code>char*</code>
Description	Returns the currently selected text of component obj .	
Targets	Textarea, Textfield	

getstate

Synopsis	<code>int j_getstate (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns J_TRUE , if component is selected, J_FALSE otherwise.
Targets	Checkbox, Radiobutton, Checkmenuitem

gettext

Synopsis	<code>char* j_gettext (int obj , char* str);</code>				
Arguments	<table> <tr> <td><code>obj</code></td><td><code>int</code></td></tr> <tr> <td><code>str</code></td><td><code>char*</code></td></tr> </table>	<code>obj</code>	<code>int</code>	<code>str</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>str</code>	<code>char*</code>				
Description	returns the component 's text or label.				
Targets	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield				
Example	<pre> char str[256]; : label = j_label(frame,"Hello World"); printf("%s",j_gettext(label,str)); : > Hello World </pre>				

getvalue

Synopsis	int j_getvalue (int obj);
Arguments	obj int
Description	Returns the current setting of the scrollbar.
Targets	Scrollbar

getviewportheight

Synopsis	int j_getviewportheight (int obj);
Arguments	obj int
Description	Returns the height of the component 's obj port (the area that is shown)
Targets	Scrollpane

getviewportwidth

Synopsis	<code>int j_getviewportwidth (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns the width of the component 's obj port (the area that is shown)
Targets	Scrollpane

getwidth

Synopsis `int j_getwidth (int obj);`

Arguments `obj` `int`

Description Returns the width of component **obj**.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Image

Example

```
:
label = j_getlabel(frame,"Hello World");
printf("%s",j_getwidth(label));
:

> 84
```

getxpos

Synopsis	int j_getxpos (int obj);
Arguments	obj int
Description	Returns the current horizontal position of component obj in its parent's coordinate space.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

getypos

Synopsis	int j_getypos (int obj);
Arguments	obj int
Description	Returns the current vertical position of component obj in its parent's coordinate space.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

graphicbutton

Synopsis	<code>int j_graphicbutton (int obj , char* filename);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>filename</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>filename</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>filename</code>	<code>char*</code>				
Description	Creates a new <code>graphicbutton</code> component with the image loaded from filename and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: frame = j_frame("j_graphicbutton"); button = j_graphicbutton(frame,"save.gif"); :</pre>				



graphiclabel

Synopsis	<code>int j_graphiclabel (int obj , char* str);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>str</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>str</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>str</code>	<code>char*</code>				
Description	Creates a new <code>graphiclabel</code> component with the image loaded from filename and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: frame = j_frame("j_graphiclabel"); label = j_graphiclabel(frame,"new.gif"); :</pre>				

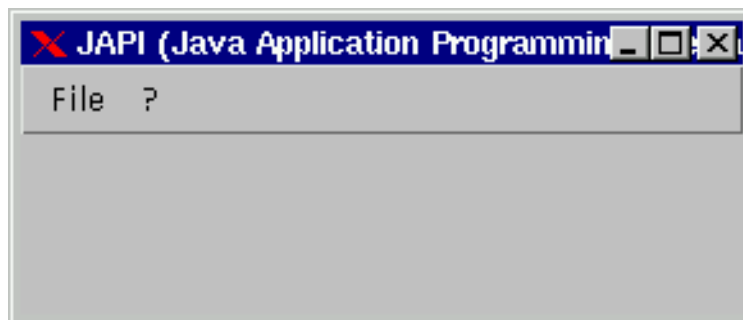


hasfocus

Synopsis	int j_hasfocus (int obj);
Arguments	obj int
Description	Returns J_TRUE if the component has the focus, J_FALSE otherwise.
Targets	Focuslistener

helpmenu

Synopsis	<code>int j_helpmenu (int obj , char* label);</code>
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>label</div> <div>char*</div> </div>
Description	Creates a new helpmenu component with the specified label and returns its event number.
Targets	Menubar
Example	<pre> : frame = j_frame("Menu Komponenten"); menubar = j_menubar(frame); file= j_menu(menubar,"File"); help= j_helpmenu(menubar,"?"); : </pre>



hide

Synopsis	<code>void j_hide (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Hides the component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

hscrollbar

Synopsis `int j_hscrollbar (int obj);`

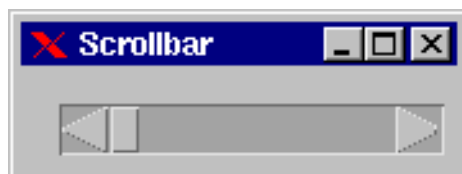
Arguments `obj` `int`

Description Creates a new horizontal scrollbar and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame, Scrollpane

Example

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



image

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

Arguments width int
 height int

Description Creates a new (memory) image component with the given **width** and **height** and returns its event number. The return value is the eventnumber of the image. On error -1 will be returned.

Example

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

insert

Synopsis	int j_insert (int obj , int pos , char* label);
Arguments	obj int pos int label char*
Description	inserts a new item to component obj at position pos with the specified label .
Targets	List, Choice

inserttext

Synopsis	<code>void j_inserttext (int obj , char* text , int pos);</code>	
Arguments	<code>obj</code>	<code>int</code>
	<code>text</code>	<code>char*</code>
	<code>pos</code>	<code>int</code>
Description	Places additional text within the component at the given position pos .	
Targets	Textarea	

isparent

Synopsis	int j_isparent (int obj , int cont);
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>cont</div> <div>int</div> </div>
Description	Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

isresizable

Synopsis	int j_isresizable (int obj);
Arguments	obj int
Description	returns true if component is resizable, false otherwise
Targets	Dialog, Frame

isselect

Synopsis	int j_isselect (int obj , int item);	
Arguments	obj	int
	item	int
Description	Returns J_TRUE if the particular item is currently selected, J_FALSE otherwise.	
Targets	List	

isvisible

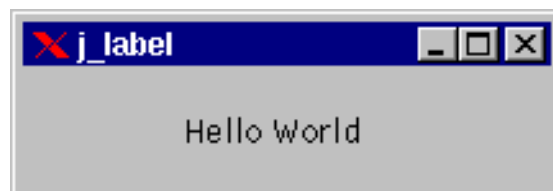
Synopsis	<code>int j_isvisible (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Returns J_TRUE if obj is visible, J_FALSE otherwise.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

keylistener

Synopsis	<code>int j_keylistener (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Adds a new key listener to component obj , and returns its event number.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

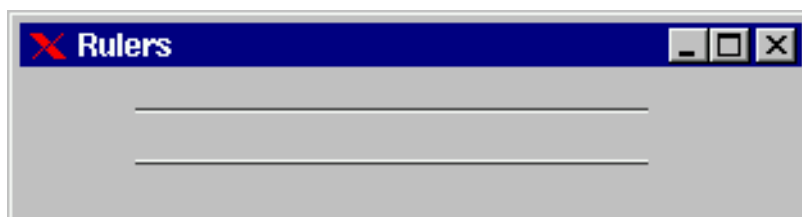
label

Synopsis	<code>int j_label (int obj , char* label);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>label</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>label</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>label</code>	<code>char*</code>				
Description	Creates a new label component with the specified label and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: frame = j_frame("j_label"); label = j_label(frame,"Hello World"); :</pre>				



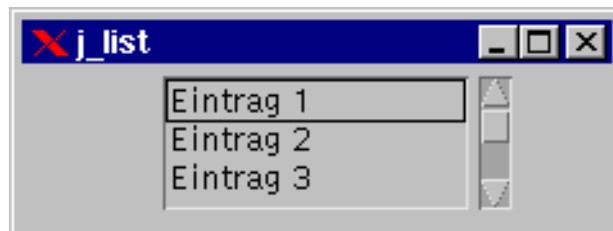
line

Synopsis	<code>int j_line (int obj , int orient , int style , int length);</code>
Arguments	<div> <div>obj</div> <div>orient</div> <div>style</div> <div>length</div> </div> <div> <div>int</div> <div>int</div> <div>int</div> <div>int</div> </div>
Description	<p>Creates a new line component with the specified length and returns its event number. A line may be used to separate groups of components. On Error <code>-1</code> will returned. The parameter orient specifies the orientation of the line:</p> <ul style="list-style-type: none"> • J_HORIZONTAL : horizontal line • J_VERTICAL : vertical line <p>The Parameter style specifies the linestyle:</p> <ul style="list-style-type: none"> • J_LINEDOWN : etched-in linestyle. • J_LINEUP : etchet-out linestyle.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre> : line1 = j_line(frame,J_HORIZONTAL,J_LINEDOWN,200); line2 = j_line(frame,J_HORIZONTAL,J_LINEUP,200); : </pre>



list

Synopsis	<code>int j_list (int obj , int rows);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>rows</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>rows</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>rows</code>	<code>int</code>				
Description	Creates a new list component with the specified number of rows and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: list = j_list(frame,3); j_additem(list,"Eintrag 1"); j_additem(list,"Eintrag 2"); :</pre>				

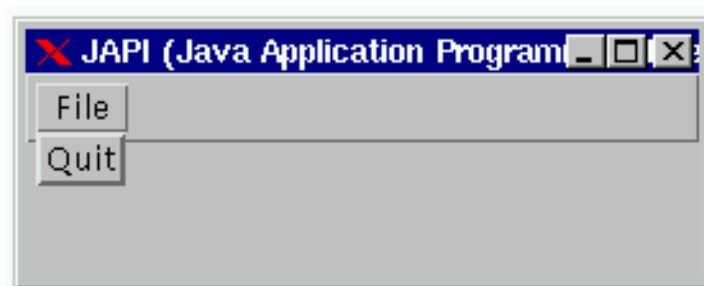


loadimage

Synopsis	<code>int j_loadimage (char* filename);</code>
Arguments	<code>filename</code> <code>char*</code>
Description	<p>Loads the Image from file filename and returns its eventnumber. The file could be of the following format:</p> <ul style="list-style-type: none">• GIF• JPEG• BMP• PPM
Example	<pre>: image = j_loadimage("mandel.jpg"); :</pre>

menu

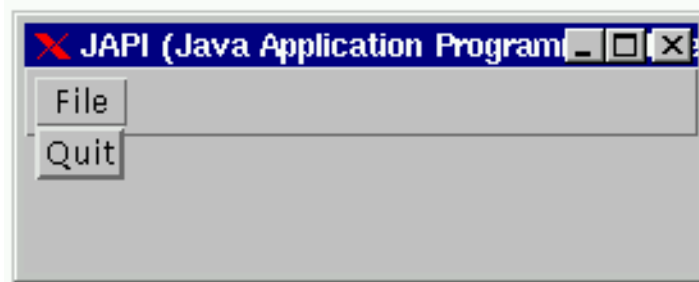
Synopsis	<code>int j_menu (int obj , char* str);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>str</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>str</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>str</code>	<code>char*</code>				
Description	Creates a new menu component with the specified label and returns its event number.				
Targets	Menubar, Menu				
Example	<pre>: frame = j_frame("Menu Komponenten"); menubar = j_menubar(frame); file = j_menu(menubar,"File"); quit = j_menuitem(file,"Quit"); :</pre>				



menubar

Synopsis	<code>int j_menubar (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Creates a new menubar and returns its event number.
Targets	Frame
Example	

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



menuitem

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

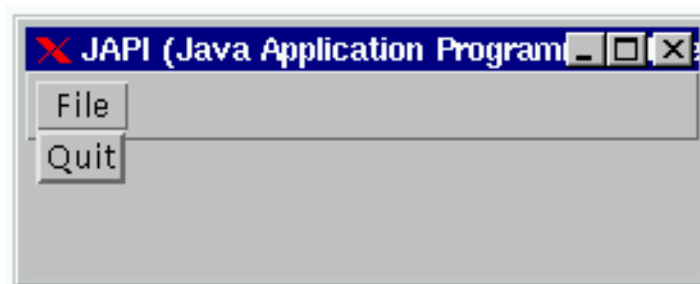
Arguments `obj` `int`
 `label` `char*`

Description Creates a new menuitem with the specified **label** and returns its event number.

Targets Menu, Popupmenu, Helpmenu

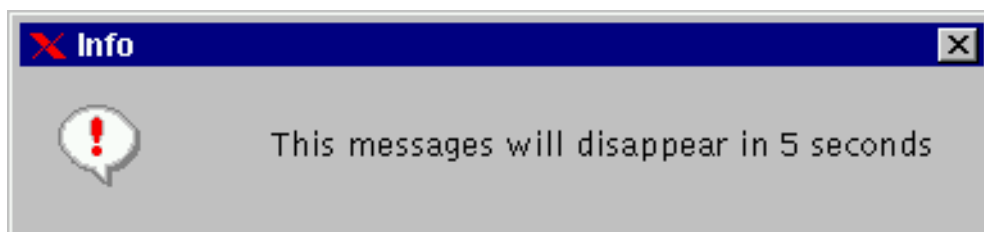
Example

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



messagebox

Synopsis	<code>void j_messagebox (int obj , char* title , char* text);</code>	
Arguments	obj	int
	title	char*
	text	char*
Description	Shows a messagebox with the specified title and text and returns its event number. In the case of error <code>-1</code> will be returned. A MessageBox generates an event, if the close icon is clicked.	
Targets	Frame	
Example	<pre> : mbox = j_messagebox(frame,"Info","This messages will disappear in 5 seconds"); j_sleep(5000); j_dispose(mbox); : </pre>	



mouselistener

Synopsis	int j_mouselistener (int obj , int kind);				
Arguments	<table><tr><td>obj</td><td>int</td></tr><tr><td>kind</td><td>int</td></tr></table>	obj	int	kind	int
obj	int				
kind	int				
Description	<p>Adds a new mouse listener to component obj, and returns its event number. An event occurs, if the user action is of kind kind. Possible values for kind:</p> <ul style="list-style-type: none">• J_ENTERED : An event occurs if the mouse cursor has been moved into the component obj.• J_MOVED : An event occurs if the mouse cursor has been moved inside the component obj.• J_EXITED : An event occurs if the mouse cursor has been moved out of the component obj.• J_PRESSED : An event occurs if a mouse button was pressed.• J_DRAGGED : An event occurs if the mouse cursor has been dragged (moved with pressed button) inside the component obj.• J_RELEASED : An event occurs if a mouse button was released.• J_DOUBLECLICK : An event occurs if a mouse button was doubleclicked.				
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield				

multiplemode

Synopsis	int j_multiplemode (int obj , int bool);	
Arguments	obj	int
	bool	int
Description	if bool is J_TRUE , selection mode is turned to multiplemode.	
Targets	List	

nextaction

Synopsis

```
int j_nextaction ( );
```

Description

Waits for the next event.

pack

Synopsis	<code>void j_pack (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Resizes component to the minimal size of contained components.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: j_setflowlayout(jframe,J_HORIZONTAL); canvas = j_canvas(frame,200,50); j_setnamedcolorbg(canvas,J_RED); j_pack(frame); :</pre>



panel

Synopsis `int j_panel (int obj);`

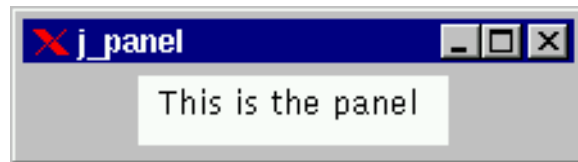
Arguments `obj` `int`

Description Creates a new panel component and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

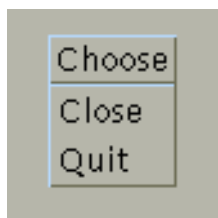
Example

```
:  
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	<code>int j_popupmenu (int obj , char* label);</code>
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>label</div> <div>char*</div> </div>
Description	Creates a new popupmenu with the specified label and returns its event number.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield
Example	<pre> : choose = j_popupmenu(frame,"Choose"); close = j_menuitem(choose,"Close"); quit = j_menuitem(choose,"Quit"); j_showpopup(choose,100,100); : </pre>



print

Synopsis `void j_print (int obj);`

Arguments `obj` `int`

Description prints the component .

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Canvas, Image, Printer

Example

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



printer

Synopsis	<code>int j_printer (int frame);</code>		
Arguments	<table><tr><td>frame</td><td>int</td></tr></table>	frame	int
frame	int		
Description	Creates a new object, representing a paper of the printer and returns its event number. On error <code>-1</code> will be returned. A printer object can be used like a canvas, where all drawing funktions will be passed to the printer, instead of a window. A printer generates no event.		
Targets	Frame		
Example	<pre>: printer = j_printer(frame); j_drawimage(printer,image,100,100); :</pre>		

quit

Synopsis

```
void j_quit ( );
```

Description

Cancels the connection to the JAPI Kernel.

radiobutton

Synopsis	<code>int j_radiobutton (int obj , char* label);</code>
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>label</div> <div>char*</div> </div>
Description	Creates a new radiobutton with the specified label and returns its event number.
Targets	Radiogroup
Example	<pre> : radiogroup = j_radiogroup(frame); radio1 = j_radiobutton(radiogroup,"Radiobutton 1"); radio2 = j_radiobutton(radiogroup,"Radiobutton 2"); : </pre>



radiogroup

Synopsis `int j_radiogroup (int obj);`

Arguments `obj` `int`

Description Creates a new radiogroup and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



random

Synopsis

```
int j_random ( );
```

Description

Generates a pseudo random number. The returned value will be in the range of 0 to 2147483647 ($2^{31} - 1$).

release

Synopsis	<code>void j_release (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Releases component obj from its parent component (container).
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

releaseall

Synopsis	<code>void j_releaseall (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Releases all components from component obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame

remove

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

Arguments	<code>obj</code>	<code>int</code>
	<code>item</code>	<code>int</code>

Description removes the Item with the Index **item** from the component .

Targets List, Choice

removeall

Synopsis	<code>int j_removeall (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Removes all items from the component .
Targets	List, Choice

removeitem

Synopsis	int j_removeitem (int obj , char* item);	
Arguments	obj	int
	item	char*
Description	remove the first occurrence of item from the component .	
Targets	List, Choice	

replacetext

Synopsis	<code>void j_replacetext (int obj , char* text , int start , int end);</code>	
Arguments	obj	int
	text	char*
	start	int
	end	int
Description	Replaces the text from starting position start to ending position end with the given text .	
Targets	Textarea	

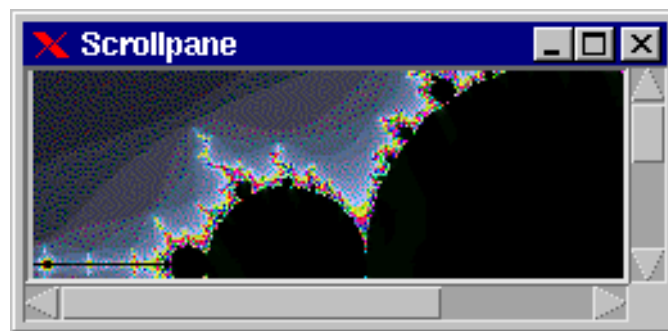
saveimage

Synopsis	<code>int j_saveimage (int obj , char* filename , int filetype);</code>						
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>filename</code></td><td><code>char*</code></td></tr><tr><td><code>filetyp</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>filename</code>	<code>char*</code>	<code>filetyp</code>	<code>int</code>
<code>obj</code>	<code>int</code>						
<code>filename</code>	<code>char*</code>						
<code>filetyp</code>	<code>int</code>						
Description	<p>Saves the components image to file filename. The specified file format can be:</p> <ul style="list-style-type: none">• J_BMP Win32 Bitmap Format• J_PPM Portable pixmap						
Example	<pre>: if(! j_saveimage(canvas,"mandel.bmp",J_BMP)) printf("Error saving Bitmap file\n"); :</pre>						

scrollpane

Synopsis	int j_scrollpane (int obj);
Arguments	obj int
Description	Creates a new scrollpane component and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	

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



select

Synopsis	int j_select (int obj , int item);	
Arguments	obj	int
	item	int
Description	Makes the given item the selected one for the component .	
Targets	List, Choice	

selectall

Synopsis	<code>void j_selectall (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Selects all the text in the component .
Targets	Textarea, Textfield

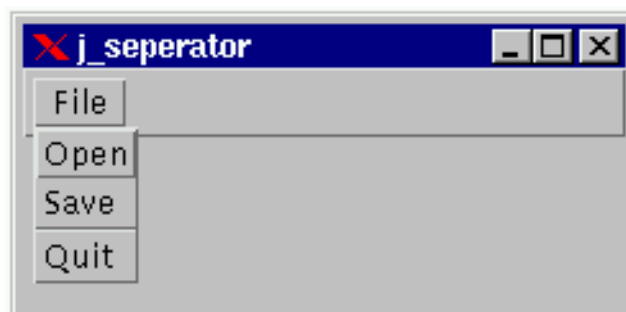
selecttext

Synopsis	void j_selecttext (int obj , int start , int end);	
Arguments	obj	int
	start	int
	end	int
Description	Selects text from starting position start to ending position end .	
Targets	Textarea, Textfield	

seperator

Synopsis	<code>void j_seperator (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Adds a separator bar to the component .
Targets	Menu, HelpMenu, Popupmenu
Example	

```
:  
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	<code>void j_setalign (int obj , int align);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>align</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>align</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>align</code>	<code>int</code>				
Description	Sets the alignment in component obj to align . Needs a flowlayout Manager.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				

setblockinc

Synopsis	int j_setblockinc (int obj , int val);	
Arguments	obj	int
	val	int
Description	Changes the block increment amount for the component to val .	
Targets	Scrollbar	

setborderlayout

Synopsis	<code>void j_setborderlayout (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Adds a borderlayout manager to component obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame

setborderpos

Synopsis	void j_setborderpos (int obj , int pos);	
Arguments	obj	int
	pos	int
Description	Moves component obj at a certain position. The outer container needs a border layout manager.	
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield	

setcolor

Synopsis `void j_setcolor (int obj , int r , int g , , int b);`

Arguments

<code>obj</code>	<code>int</code>
<code>r</code>	<code>int</code>
<code>g,</code>	<code>int</code>
<code>b</code>	<code>int</code>

Description Sets the foreground color to the (**r**, **g**, **b**) values.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

Example

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



setcolorbg

Synopsis	<code>void j_setcolorbg (int obj , int r , int g , int b);</code>
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>r</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>g,</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>b</div> <div>int</div> </div>
Description	Sets the background color to the (r , g , b) values.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield
Example	<pre> : button = j_button(frame,"Hello World"); j_setcolorbg(button,150,0,0); j_settext(button,"Hello World"); : </pre>



setcolumns

Synopsis

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

Arguments

obj	int
columns	int

Description

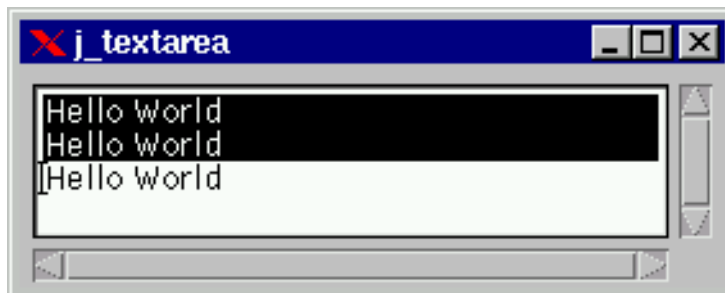
Sets the number of columns for **obj** to **columns**.

Targets

Textarea, Textfield, Gridlayout

Example

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



setcurpos

Synopsis	void j_setcurpos (int obj , int pos);	
Arguments	obj	int
	pos	int
Description	Change the location of the text cursor to the specified position pos .	
Targets	Textarea, Textfield	

setcursor

Synopsis	int j_setcursor (int obj , int cursor);				
Arguments	<table><tr><td>obj</td><td>int</td></tr><tr><td>cursor</td><td>int</td></tr></table>	obj	int	cursor	int
obj	int				
cursor	int				
Description	Changes the component 's obj cursor to the specified cursor .				
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield				

setdebug

Synopsis `void j_setdebug (int level);`

Arguments `level` `int`

Description Sets the debuglevel to **level**.

setechochar

Synopsis	void j_setechochar (int obj , char chr);	
Arguments	obj	int
	chr	char
Description	Changes the character chr that is used to echo all user input in the component .	
Targets	Textfield	

seteditable

Synopsis	void j_seteditable (int obj , int bool);
Arguments	obj int bool int
Description	Allows to make the component editable (bool =J_TRUE) or read-only (bool =J_FALSE).
Targets	Textarea, Textfield

setfixlayout

Synopsis	<code>void j_setfixlayout (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Adds a fixlayout manager to component obj (default layout manager).
Targets	Panel, Borderpanel, Window, Dialog, Frame

setflowfill

Synopsis	<code>void j_setflowfill (int obj , int bool);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>bool</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>bool</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>bool</code>	<code>int</code>				
Description	Resizes all containing component to the height (width) of component obj . Needs a flowlayout manager.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				

setflowlayout

Synopsis	void j_setflowlayout (int obj , int align);	
Arguments	obj	int
	align	int
Description	Adds a flowlayout manager to component obj with the specified alignment .	
Targets	Panel, Borderpanel, Window, Dialog, Frame	

setfocus

Synopsis	int j_setfocus (int obj);
Arguments	obj int
Description	Directs the input focus to component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

setfont

Synopsis	<code>void j_setfont (int obj , int name , int style , int size);</code>								
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>name</code></td><td><code>int</code></td></tr><tr><td><code>style</code></td><td><code>int</code></td></tr><tr><td><code>size</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>name</code>	<code>int</code>	<code>style</code>	<code>int</code>	<code>size</code>	<code>int</code>
<code>obj</code>	<code>int</code>								
<code>name</code>	<code>int</code>								
<code>style</code>	<code>int</code>								
<code>size</code>	<code>int</code>								
Description	Changes the font to the given characteristics name , style and size .								
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, MenuItem, Check-BoxMenuItem, Menu, HelpMenu, Popupmenu								
Example	<pre>: label = j_label(jframe,"Hello World"); j_setfont(label,J_TIMES,J_PLAIN,18); :</pre>								



setfontname

Synopsis `void j_setfontname (int obj , int name);`

Arguments `obj` `int`
 `name` `int`

Description Changes the font to the given **name**.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, MenuItem, Check-BoxMenuItem, Menu, HelpMenu, Popupmenu

Example

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



setfontsize

Synopsis `void j_setfontsize (int obj , int size);`

Arguments `obj` `int`
 `size` `int`

Description Changes the font to the given **size**.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, MenuItem, Check-BoxMenuItem, Menu, HelpMenu, Popupmenu

Example

```
:  
label = j_label(jframe,"Hello World");  
j_setfontsize(label,24);  
:
```



setfontstyle

Synopsis `void j_setfontstyle (int obj , int style);`

Arguments `obj` `int`
 `style` `int`

Description Changes the font to the given **style**.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, MenuItem, Check-BoxMenuItem, Menu, HelpMenu, Popupmenu

Example

```
:  
label = j_label(jframe,"Hello World");  
j_setfontstyle(label,J_BOLD+J_ITALIC);  
:
```



setgridlayout

Synopsis	void j_setgridlayout (int obj , int row , int col);	
Arguments	obj	int
	row	int
	col	int
Description	Adds a gridlayout manager to component obj with the specified rows and columns .	
Targets	Panel, Borderpanel, Window, Dialog, Frame	

sethgap

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

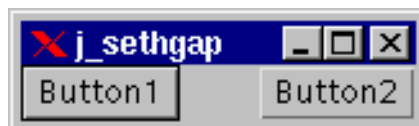
Arguments `obj` `int`
 `hgap` `int`

Description Sets the horizontal gap between components to **hgap** Pixel.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



seticon

Synopsis	<code>void j_seticon (int frame , int icon);</code>				
Arguments	<table><tr><td><code>frame</code></td><td><code>int</code></td></tr><tr><td><code>icon</code></td><td><code>int</code></td></tr></table>	<code>frame</code>	<code>int</code>	<code>icon</code>	<code>int</code>
<code>frame</code>	<code>int</code>				
<code>icon</code>	<code>int</code>				
Description	Sets the image icon to display when the frame is iconized. Not all platforms support the concept of iconizing a window.				
Targets	Frame				
Example	<pre>: frame = j_frame("Hello World"); j_seticon(frame,j_loadimage("icon.gif")); :</pre>				

setimage

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

Arguments `obj` `int`
 `image` `int`

Description Sets the **image** to be displayed in **obj**.

Targets Graphicbutton, Graphiclabel

Example

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



setinsets

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

Arguments obj int
 top int
 bottom int
 left int
 right int

Description Set the insets to the specified values.

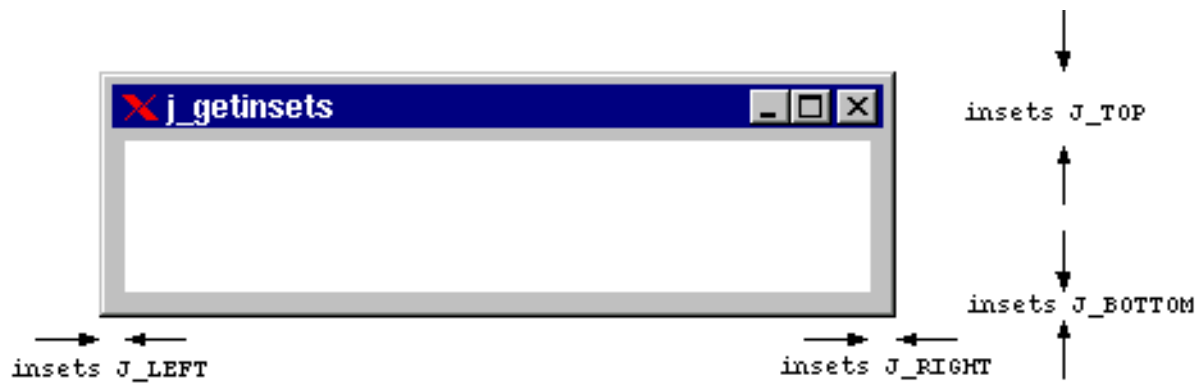
Targets Panel, Borderpanel, Window, Dialog, Frame

Example

```

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

```



setmax

Synopsis	int j_setmax (int obj , int val);	
Arguments	obj	int
	val	int
Description	Changes the maximum value for the component to val .	
Targets	Scrollbar	

setmin

Synopsis	int j_setmin (int obj , int val);	
Arguments	obj	int
	val	int
Description	Changes the minimum value for the component to val .	
Targets	Scrollbar	

setnamedcolor

Synopsis	void j_setnamedcolor (int obj , int color);	
Arguments	obj	int
	color	int
Description	Sets the foreground color to a predefined color .	
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield	

setnamedcolorbg

Synopsis	<code>void j_setnamedcolorbg (int obj , int color);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>color</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>color</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>color</code>	<code>int</code>				
Description	Sets the background color to a predefined color .				
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield				

setnolayout

Synopsis	<code>void j_setnolayout (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Removes the current layout manager from component obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame

setpos

Synopsis	<code>void j_setpos (int obj , int xpos , int ypos);</code>						
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>xpos</code></td><td><code>int</code></td></tr><tr><td><code>ypos</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>xpos</code>	<code>int</code>	<code>ypos</code>	<code>int</code>
<code>obj</code>	<code>int</code>						
<code>xpos</code>	<code>int</code>						
<code>ypos</code>	<code>int</code>						
Description	Relocates the component obj to the specified Position (xpos,ypos).						
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield						

setradiogroup

Synopsis	int j_setradiogroup (int rbutton, , int rgroup);				
Arguments	<table><tr><td>rbutton,</td><td>int</td></tr><tr><td>rgroup</td><td>int</td></tr></table>	rbutton,	int	rgroup	int
rbutton,	int				
rgroup	int				
Description	Sets radiobuttons rbutton group to be the specified radiogroup rgroup . If the radiobuttons is already in a different radiogroup, it is first taken out of that group.				
Targets	Radiobutton				

setresizable

Synopsis	<code>void j_setresizable (int obj , int resizable);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>resizable</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>resizable</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>resizable</code>	<code>int</code>				
Description	The component cannot be resized, if resizable is <code>J_FALSE</code> .				
Targets	Dialog, Frame				
Example	<pre>: frame = j_frame("fixsized Frame"); j_setreziisable(frame,J_FALSE); :</pre>				

setrows

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

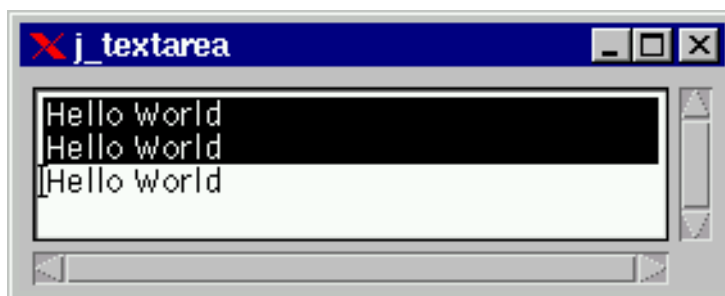
Arguments `obj` `int`
 `rows` `int`

Description Sets the number of rows for **obj** to **rows**.

Targets Textarea, Gridlayout

Example

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

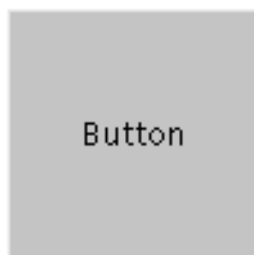


setshortcut

Synopsis	void j_setshortcut (int obj , char chr);	
Arguments	obj	int
	chr	char
Description	Changes the shortcut chr of the component .	
Targets	MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu	

setsize

Synopsis	<code>void j_setsize (int obj , int width , int height);</code>
Arguments	<div style="display: flex; justify-content: space-between;"> <div>obj</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>width</div> <div>int</div> </div> <div style="display: flex; justify-content: space-between;"> <div>height</div> <div>int</div> </div>
Description	Resizes component obj to specified width and height .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield
Example	<pre> : button = j_button(frame,"Button"); j_setsize(button,100,100); : </pre>



setslidesize

Synopsis `int j_setslidesize (int obj , int val);`

Arguments	<code>obj</code>	<code>int</code>
	<code>val</code>	<code>int</code>

Description Changes the slide size to **val**.


Targets Scrollbar

setstate

Synopsis	<code>void j_setstate (int obj , int bool);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>bool</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>bool</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>bool</code>	<code>int</code>				
Description	The component becomes selected, if bool is <code>J_TRUE</code> .				
Targets	Checkbox, Radiobutton, Checkmenuitem				

settext

Synopsis	<code>void j_settext (int obj , char* str);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>str</code></td><td><code>char*</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>str</code>	<code>char*</code>
<code>obj</code>	<code>int</code>				
<code>str</code>	<code>char*</code>				
Description	Sets the content or the label of the component obj to str .				
Targets	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield				
Example	<pre>: button = j_button(frame,"Hello World"); j_settext(button,"Goodbye"); :</pre>				



setunitinc

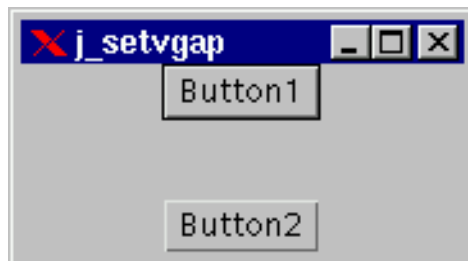
Synopsis	int j_setunitinc (int obj , int val);	
Arguments	obj	int
	val	int
Description	Changes the unit increment amount for the component to val	
Targets	Scrollbar	

setvalue

Synopsis	void j_setvalue (int obj , int val);	
Arguments	obj	int
	val	int
Description	Changes the current value of the component to val .	
Targets	Scrollbar	

setvgap

Synopsis	<code>void j_setvgap (int obj , int vgap);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>vgap</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>vgap</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>vgap</code>	<code>int</code>				
Description	Sets the vertical gap between components to vgap Pixel.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: j_setflowlayout(frame,J_VERTICAL); button1 = j_button(frame,"Button1"); button2 = j_button(frame,"Button2"); j_setvgap(frame,30); :</pre>				



setxor

Synopsis	<code>void j_setxor (int obj , int bool);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>bool</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>bool</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>bool</code>	<code>int</code>				
Description	Changes painting mode to XOR mode, if <code>bool = J_TRUE</code> . In this mode, drawing the same object in the same color at the same location twice has no net effect.				
Targets	Canvas, Image, Printer				

show

Synopsis	<code>void j_show (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Shows the component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield

showpopup

Synopsis	void j_showpopup (int obj , int xpos , int ypos);	
Arguments	obj	int
	xpos	int
	ypos	int
Description	Shows the component at specified Position (xpos,ypos).	
Targets	Popupmenu	

sleep

Synopsis `int j_sleep (int msec);`

Arguments `msec int`

Description `Suspends the execution for msec milliseconds.`

start

Synopsis `int j_start ();`

Description Get in touch with a running japi kernel or start a neu one.

Example

```

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


sync

Synopsis `void j_sync ();`

Description Synchronizes the application with the JAPI kernel.

textarea

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

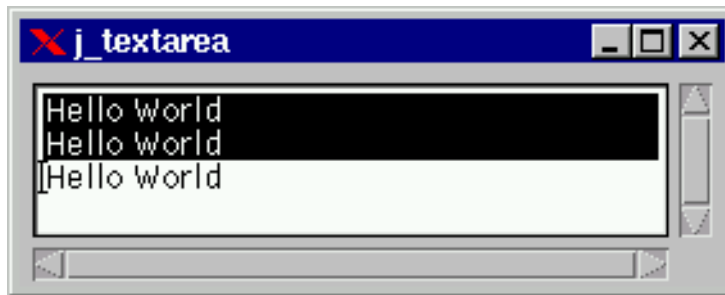
Arguments `obj` `int`
 `rows` `int`
 `columns` `int`

Description Creates a new textarea component with the specified number of **rows** **columns** and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

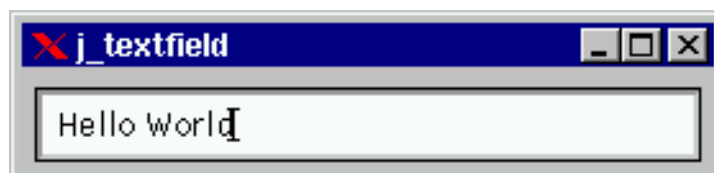
Example

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



textfield

Synopsis	<code>int j_textfield (int obj , int columns);</code>				
Arguments	<table><tr><td><code>obj</code></td><td><code>int</code></td></tr><tr><td><code>columns</code></td><td><code>int</code></td></tr></table>	<code>obj</code>	<code>int</code>	<code>columns</code>	<code>int</code>
<code>obj</code>	<code>int</code>				
<code>columns</code>	<code>int</code>				
Description	Creates a new textfield component with the specified number of columns and returns its event number.				
Targets	Panel, Borderpanel, Window, Dialog, Frame				
Example	<pre>: frame = j_frame("j_textfield"); text = j_textfield(frame,30) :</pre>				



translate

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

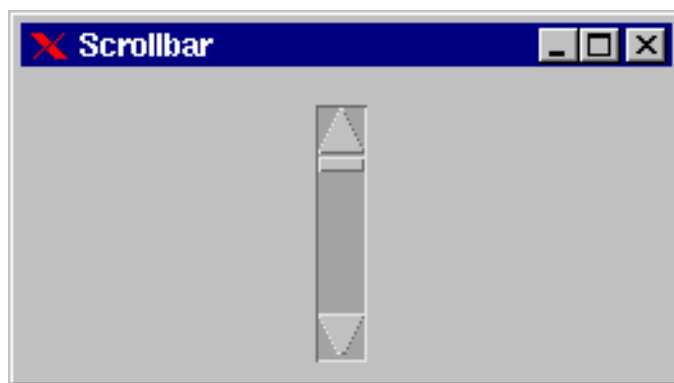
Arguments `obj int`
 `x int`
 `y int`

Description Moves the origin of drawing operations to (**x**, **y**).

Targets Canvas, Image, Printer

vscrollbar

Synopsis	<code>int j_vscrollbar (int obj);</code>
Arguments	<code>obj</code> <code>int</code>
Description	Creates a new vertical scrollbar and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame, Scrollpane
Example	<pre>: scroll=j_vscrollbar(frame); j_setpos(scroll,120,40); j_setsize(scroll,20,100); :</pre>



window

Synopsis `int j_window (int obj);`

Arguments `obj` `int`

Description Creates a new simple window and returns its event number.

Targets `Frame`

Example

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

Mouse pressed at 108:179

windowlistener

Synopsis	<code>int j_windowlistener (int window , int kind);</code>
Arguments	<div style="display: flex; justify-content: space-between;"> <div> window kind </div> <div> int int </div> </div>
Description	<p>Adds a new windowlistener to component obj, and returns its event number. An event occurs, if the user action is of kind kind. Possible values for kind:</p> <ul style="list-style-type: none"> • J_ACTIVATED : An event occurs when the component is activated. • J_DEACTIVATED : An event occurs when the component is deactivated. • J_OPENED : An event occurs when the component has been opened. • J_CLOSED : An event occurs when the component has been closed. • J_ICONFIED : An event occurs when the component is iconfied. • J_DEICONFIED : An event occurs when the component is deiconfied. • J_CLOSING : An event occurs when the close icon has been clicked .
Targets	Window, Dialog, Frame