

japi

Reference Manual

Merten Joost

Inhaltsverzeichnis

I	Reference	5
1	Components	7
	Button	7
	Borderpanel	10
	Canvas	15
	Checkbox	20
	CheckmenuItem	23
	Choice	24
	Dialog	27
	Focuslistener	33
	Frame	34
	Helpmenu	40
	Hscrollbar	41
	Graphicbutton	44
	Graphiclabel	47
	Image	50
	KeyListener	53
	Label	54
	Led	57
	List	60
	Menu	64
	MenuItem	66
	Meter	67
	Mouselistener	70
	Panel	71
	PopupMenu	76
	Printer	78
	Progressbar	80
	Radiobutton	83
	Sevensegment	86
	ScrollPane	89
	Textarea	92
	Textfield	96
	Vscrollbar	100
	Window	103

2 Functions	109
additem	109
add	110
alertbox	111
appendtext	112
beep	113
borderpanel	114
button	115
canvas	116
checkbox	117
checkmenuItem	118
choicebox2	119
choicebox3	120
choice	121
cliprect	122
componentlistener	123
connect	124
delete	125
deselect	126
dialog	127
disable	128
dispose	129
drawarc	130
drawcircle	131
drawimagesource	132
drawimage	133
drawline	134
drawoval	135
drawpixel	136
drawpolygon	137
drawpolyline	138
drawrect	139
drawroundrect	140
drawscaledimage	141
drawstring	142
enable	143
filedialog	144
fileselect	145
fillarc	146
fillcircle	147
filloval	148
fillpolygon	149
fillrect	150
fillroundrect	151
focuslistener	152
frame	153
getaction	154

getcolumns	155
getcurpos	156
getdanger	157
getfontascent	158
getfontheight	159
getheight	160
getimagesource	161
getimage	162
getinheight	163
getinsets	164
getinwidth	165
getitemcount	166
getitem	167
getkeychar	168
getkeycode	169
getlayoutid	170
getlength	171
getmousebutton	172
getmousex	173
getmousey	174
getparentid	175
getparent	176
getrows	177
getscaledimage	178
getscreenheight	179
getscreenwidth	180
getselect	181
getselend	182
getselstart	183
getseltext	184
getstate	185
getstringwidth	186
gettext	187
getvalue	188
getviewportheight	189
getviewportwidth	190
getwidth	191
getxpos	192
getypos	193
graphicbutton	194
graphiclabel	195
hasfocus	196
helpmenu	197
hide	198
hscrollbar	199
image	200
insert	201

inserttext	202
isparent	203
isresizable	204
isselect	205
isvisible	206
keylistener	207
label	208
led	209
line	210
list	211
loadimage	212
menubar	213
menuItem	214
menu	215
messagebox	216
meter	217
mouselistener	218
multiplemode	219
nextaction	220
pack	221
panel	222
popupmenu	223
printer	224
print	225
progressbar	226
quit	227
radiobutton	228
radiogroup	229
random	230
releaseall	231
release	232
removeall	233
removeitem	234
remove	235
replacetext	236
saveimage	237
scrollpane	238
selectall	239
select	240
selecttext	241
separator	242
setalign	243
setblockinc	244
setborderlayout	245
setborderpos	246
setcolorbg	247
setcolor	248

setcolumns	249
setcurpos	250
setcursor	251
setdanger	252
setdebug	253
setechochar	254
seteditable	255
setfixlayout	256
setflowfill	257
setflowlayout	258
setfocus	259
setfontname	260
setfontsize	261
setfontstyle	262
setFont	263
setgridlayout	264
sethgap	265
seticon	266
setimage	267
setinsets	268
setMax	269
setmin	270
setnamedcolorbg	271
setnamedcolor	272
setnolayout	273
start	274
setpos	275
setradiogroup	276
setresizable	277
setrows	278
setshortcut	279
setsize	280
setslidesize	281
setstate	282
settext	283
setunitinc	284
setValue	285
setvgap	286
setxor	287
sevensegment	288
showpopup	289
show	290
sleep	291
start	292
sync	293
textarea	294
textfield	295

translate	296
vscrollbar	297
windowlistener	298
window	299

Teil I

Reference

Kapitel 1

Components

Button

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

	Returns the height of button obj .
j_getlength	<i>function j_getlength (obj)</i> Returns the length of button 's label or text.
j_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of button obj .
j_gettext	<i>function\$ j_gettext (obj)</i> returns the button 's text or label.
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of button obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of button obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of button obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj)</i> Hides the button obj .
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to button obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj , 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>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the button .

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

Borderpanel

- j_borderpanel** *function j_borderpanel (obj , type)*
Creates a new borderpanel component with the style **type** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds borderpanel **obj** to container **cont**
- j_borderpanel** *function j_borderpanel (obj , type)*
Creates a new borderpanel component with the style **type** and returns its event number.
- j_button** *function j_button (obj , label\$)*
Creates a new button component with the specified **label** and returns its event number.
- j_canvas** *function j_canvas (obj , width , height)*
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** *function j_checkbox (obj , label\$)*
Creates a new checkbox component with the specified **label** and returns its event number.
- j_choice** *function j_choice (obj)*
Creates a new choice component and returns its event number.
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables borderpanel **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the borderpanel **obj**.
- j_enable** *procedure j_enable (obj)*
enables the borderpanel **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to borderpanel **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of borderpanel **obj**.

j_getfontheight	<i>function j_getfontheight (obj)</i> Returns the total pixel height of the actual font of borderpanel obj .
j_getheight	<i>function j_getheight (obj)</i> Returns the height of borderpanel obj .
j_getinheight	<i>function j_getinheight (cont)</i> Returns the height of the client size.
j_getinsets	<i>function j_getinsets (obj , side)</i> Returns the width of the specified inset.
j_getinwidth	<i>function j_getinwidth (cont)</i> Returns the width of the client size.
j_getlayoutid	<i>function j_getlayoutid (obj)</i> Returns the event number of the layoutmanager for containers obj .
j_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of borderpanel obj .
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of borderpanel obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of borderpanel obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of borderpanel obj in its parent's coordinate space.
j_graphicbutton	<i>function j_graphicbutton (obj , filename\$)</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>function j_graphiclabel (obj , str\$)</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>procedure j_hide (obj)</i> Hides the borderpanel obj .
j_hscrollbar	<i>function j_hscrollbar (obj)</i> Creates a new horizontal scrollbar and returns its event number.

j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to borderpanel obj , and returns its event number.
j_label	<i>function j_label (obj , label\$)</i> Creates a new label component with the specified label and returns its event number.
j_led	<i>function j_led (obj , style , color)</i> Creates a new led component with the specified style and the specified color color .
j_line	<i>function j_line (obj , orient , style , length)</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>function j_list (obj , rows)</i> Creates a new list component with the specified number of rows and returns its event number.
j_meter	<i>function j_meter (obj , title\$)</i> Creates a new pointer-instrument with the specified label titel .
j_mouselistener	<i>function j_mouselistener (obj , 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>procedure j_pack (obj)</i> Resizes borderpanel to the minimal size of contained components.
j_panel	<i>function j_panel (obj)</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the borderpanel .
j_progressbar	<i>function j_progressbar (obj , orient)</i> Creates a new progressbar with the specified orientation .
j_radiogroup	<i>function j_radiogroup (obj)</i> Creates a new radiogroup and returns its event number.
j_releaseall	<i>procedure j_releaseall (obj)</i> Releases all components from borderpanel obj .

j_release	<i>procedure j_release (obj)</i> Releases borderpanel obj from its parent component (container).
j_scrollpane	<i>function j_scrollpane (obj)</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>procedure j_setalign (obj , align)</i> Sets the alignment in borderpanel obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (obj)</i> Adds a borderlayout manager to borderpanel obj .
j_setborderpos	<i>procedure j_setborderpos (obj , pos)</i> Moves borderpanel obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (obj , r , g , b)</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (obj , r , g , b)</i> Sets the foreground color to the (r , g , b) values.
j_setcursor	<i>function j_setcursor (obj , cursor)</i> Changes the borderpanel 's obj cursor to the specified cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj)</i> Adds a fixlayout manager to borderpanel obj (default layout manager).
j_setflowfill	<i>procedure j_setflowfill (obj , bool)</i> Resizes all containing component to the height (width) of borderpanel obj . Needs a flowlayout manager.
j_setflowlayout	<i>procedure j_setflowlayout (obj , align)</i> Adds a flowlayout manager to borderpanel obj with the specified alignment .
j_setfocus	<i>function j_setfocus (obj)</i> Directs the input focus to borderpanel obj .
j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (obj , row , col)</i>

	Adds a gridlayout manager to borderpanel obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (obj , hgap)</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj , top , bottom , left , right)</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj)</i> Removes the current layout manager from borderpanel obj .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the borderpanel obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes borderpanel obj to specified width and height .
j_setvgap	<i>procedure j_setvgap (obj , vgap)</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj , color)</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj)</i> Shows the borderpanel obj .
j_textarea	<i>function j_textarea (obj , rows , columns)</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>function j_textfield (obj , columns)</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj)</i> Creates a new vertical scrollbar and returns its event number.

Canvas

- j_canvas** *function j_canvas (obj , width , height)*
 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** *procedure j_add (obj , cont)*
 Adds canvas **obj** to container **cont**
- j_cliprect** *procedure j_cliprect (obj , x , y , width , height)*
 Changes current clipping region to the specified rectangle (**x**, **y**, **width**, **height**).
- j_componentlistener** *function j_componentlistener (obj , kind)*
 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** *procedure j_disable (obj)*
 Disables canvas **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
 Releases the resources of the canvas **obj**.
- j_drawarc** *procedure j_drawarc (obj , x , y , rx , ry , arc1 , arc2)*
 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** *procedure j_drawcircle (obj , x , y , r)*
 Draws an unfilled circle with center (**x**, **y**) and radius **x**.
- j_drawimage** *procedure j_drawimage (obj , image , x , y)*
 Copies the image, given by its eventnumber **image**, to position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (obj , x , y , w , h , r , g , b)*
 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** *procedure j_drawline (obj , x1 , y1 , x2 , y2)*
 Draws a line connecting (**x1**,**y1**) and (**x2**,**y2**).
- j_drawoval** *procedure j_drawoval (obj , x , y , rx , ry)*
 Draws an unfilled oval with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j_drawpixel** *procedure j_drawpixel (obj , x , y)*
 Draws a pixel at (**x**,**y**).

j_drawpolygon	<i>procedure j_drawpolygon (obj , len , x , y)</i> Draws an unfilled polygon based on first len elements in x and y .
j_drawpolyline	<i>procedure j_drawpolyline (obj , len , x , y)</i> Draws a series of line segments based on first len elements in x and y .
j_drawrect	<i>procedure j_drawrect (obj , x , y , width , height)</i> Draws an unfilled rectangle from (x,y) of size width x height .
j_drawroundrect	<i>procedure j_drawroundrect (obj , x , y , width , height , arcx , 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>procedure j_drawscaledimage (obj , image , sx , sy , sw , sh , tx , ty , tw , 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>procedure j_drawstring (obj , x , y , str\$)</i> Draws text on screen at position (x,y) .
j_enable	<i>procedure j_enable (obj)</i> enables the canvas obj .
j_fillarc	<i>procedure j_fillarc (obj , x , y , rx , ry , arc1 , 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>procedure j_fillcircle (obj , x , y , r)</i> Draws an filled circle with center (x, y) and radius x .
j_filloval	<i>procedure j_filloval (obj , x , y , rx , ry)</i> Draws an filled oval with the center (x, y) and the horizontal radius rx and the vertical radius ry .
j_fillpolygon	<i>procedure j_fillpolygon (obj , len , x , y)</i> Draws an filled polygon based on first len elements in x and y .
j_fillrect	<i>procedure j_fillrect (obj , x , y , width , height)</i> Draws an filled rectangle from (x,y) of size width x height .
j_fillroundrect	<i>procedure j_fillroundrect (obj , x , y , width , height , arcx , 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>function j_focuslistener (obj)</i> Adds a new focus listener to canvas obj , and returns its event number.
j_getfontascent	<i>function j_getfontascent (obj)</i> Returns the ascent (space above the baseline) of the actual font of canvas obj .

j_getfontheight	<i>function j_getfontheight (obj)</i> Returns the total pixel height of the actual font of canvas obj .
j_getheight	<i>function j_getheight (obj)</i> Returns the height of canvas obj .
j_getimage	<i>function j_getimage (obj)</i> Copy the contents of canvas obj into an image and return its eventnumber.
j_getimagesource	<i>function j_getimagesource (obj , x , y , w , h , r , g , 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_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getscaledimage	<i>function j_getscaledimage (obj , x , y , sw , sh , tw , 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_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of canvas obj .
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of canvas obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of canvas obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of canvas obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj)</i> Hides the canvas obj .
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to canvas obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj , 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>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.</p>
j_print	<p><i>procedure j_print (obj)</i> prints the canvas .</p>
j_release	<p><i>procedure j_release (obj)</i> Releases canvas obj from its parent component (container).</p>
j_setborderpos	<p><i>procedure j_setborderpos (obj , pos)</i> Moves canvas obj at a certain position. The outer container needs a border layout manager.</p>
j_setcolorbg	<p><i>procedure j_setcolorbg (obj , r , g , b)</i> Sets the background color to the (r, g, b) values.</p>
j_setcolor	<p><i>procedure j_setcolor (obj , r , g , b)</i> Sets the foreground color to the (r, g, b) values.</p>
j_setcursor	<p><i>function j_setcursor (obj , cursor)</i> Changes the canvas 's obj cursor to the specified cursor.</p>
j_setfocus	<p><i>function j_setfocus (obj)</i> Directs the input focus to canvas obj.</p>
j_setfontname	<p><i>procedure j_setfontname (obj , name)</i> Changes the font to the given name.</p>
j_setfont	<p><i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name, style and size.</p>
j_setfontsize	<p><i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size.</p>
j_setfontstyle	<p><i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style.</p>
j_setnamedcolorbg	<p><i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color.</p>
j_setnamedcolor	<p><i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color.</p>
j_setpos	<p><i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the canvas obj to the specified Position (xpos,ypos).</p>
j_setsize	<p><i>procedure j_setsize (obj , width , height)</i> Resizes canvas obj to specified width and height.</p>

- j_setxor** *procedure j_setxor (obj , 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_show** *procedure j_show (obj)*
Shows the canvas **obj**.
- j_translate** *procedure j_translate (obj , x , y)*
Moves the origin of drawing operations to (**x**, **y**).

Checkbox

- j_checkbox** *function j_checkbox (obj , label\$)*
Creates a new checkbox component with the specified **label** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds checkbox **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables checkbox **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the checkbox **obj**.
- j_enable** *procedure j_enable (obj)*
enables the checkbox **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to checkbox **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of checkbox **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of checkbox **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of checkbox **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstate** *function j_getstate (obj)*
Returns **J_TRUE** , if checkbox is selected, **J_FALSE** otherwise.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of checkbox **obj**.

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

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

Checkmenuitem

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

Choice

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

	Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getselect	<i>function j_getselect (obj)</i> Returns the position of currently selected item.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of choice obj .
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of choice obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of choice obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of choice obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj)</i> Hides the choice obj .
j_insert	<i>function j_insert (obj , pos , label\$)</i> inserts a new item to choice obj at position pos with the specified label .
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to choice obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj , 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>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the choice .
j_release	<i>procedure j_release (obj)</i> Releases choice obj from its parent component (container).
j_removeall	<i>function j_removeall (obj)</i> Removes all items from the choice .
j_removeitem	<i>function j_removeitem (obj , item\$)</i>

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

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

Dialog

- j_dialog** *function j_dialog (obj , label\$)*
Creates a new dialog window with the specified **label** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds dialog **obj** to container **cont**
- j_borderpanel** *function j_borderpanel (obj , type)*
Creates a new borderpanel component with the style **type** and returns its event number.
- j_button** *function j_button (obj , label\$)*
Creates a new button component with the specified **label** and returns its event number.
- j_canvas** *function j_canvas (obj , width , height)*
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** *function j_checkbox (obj , label\$)*
Creates a new checkbox component with the specified **label** and returns its event number.
- j_choice** *function j_choice (obj)*
Creates a new choice component and returns its event number.
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables dialog **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the dialog **obj**.
- j_enable** *procedure j_enable (obj)*
enables the dialog **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to dialog **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of dialog **obj**.
- j_getfontheight** *function j_getfontheight (obj)*

	Returns the total pixel height of the actual font of dialog obj .
j_getheight	<i>function j_getheight (obj)</i> Returns the height of dialog obj .
j_getinheight	<i>function j_getinheight (cont)</i> Returns the height of the client size.
j_getinsets	<i>function j_getinsets (obj , side)</i> Returns the width of the specified inset.
j_getinwidth	<i>function j_getinwidth (cont)</i> Returns the width of the client size.
j_getlayoutid	<i>function j_getlayoutid (obj)</i> Returns the event number of the layoutmanager for containers obj .
j_getlength	<i>function j_getlength (obj)</i> Returns the length of dialog 's label or text.
j_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of dialog obj .
j_gettext	<i>function\$ j_gettext (obj)</i> returns the dialog 's text or label.
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of dialog obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of dialog obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of dialog obj in its parent's coordinate space.
j_graphicbutton	<i>function j_graphicbutton (obj , filename\$)</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>function j_graphiclabel (obj , str\$)</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.

j_hide	<i>procedure j_hide (obj)</i> Hides the dialog obj .
j_hscrollbar	<i>function j_hscrollbar (obj)</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isresizable	<i>function j_isresizable (obj)</i> returns true if dialog is resizable, false otherwise
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to dialog obj , and returns its event number.
j_label	<i>function j_label (obj , label\$)</i> Creates a new label component with the specified label and returns its event number.
j_led	<i>function j_led (obj , style , color)</i> Creates a new led component with the specified style and the specified color color .
j_line	<i>function j_line (obj , orient , style , length)</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>function j_list (obj , rows)</i> Creates a new list component with the specified number of rows and returns its event number.
j_meter	<i>function j_meter (obj , title\$)</i> Creates a new pointer-instrument with the specified label titel .
j_mouselistener	<i>function j_mouselistener (obj , 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>procedure j_pack (obj)</i> Resizes dialog to the minimal size of contained components.
j_panel	<i>function j_panel (obj)</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the dialog .

j_progressbar	<i>function j_progressbar (obj , orient)</i> Creates a new progressbar with the specified orientation .
j_radiogroup	<i>function j_radiogroup (obj)</i> Creates a new radiogroup and returns its event number.
j_releaseall	<i>procedure j_releaseall (obj)</i> Releases all components from dialog obj .
j_release	<i>procedure j_release (obj)</i> Releases dialog obj from its parent component (container).
j_scrollpane	<i>function j_scrollpane (obj)</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>procedure j_setalign (obj , align)</i> Sets the alignment in dialog obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (obj)</i> Adds a borderlayout manager to dialog obj .
j_setborderpos	<i>procedure j_setborderpos (obj , pos)</i> Moves dialog obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (obj , r , g , b)</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (obj , r , g , b)</i> Sets the foreground color to the (r , g , b) values.
j_setcursor	<i>function j_setcursor (obj , cursor)</i> Changes the dialog 's obj cursor to the specified cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj)</i> Adds a fixlayout manager to dialog obj (default layout manager).
j_setflowfill	<i>procedure j_setflowfill (obj , bool)</i> Resizes all containing component to the height (width) of dialog obj . Needs a flowlayout manager.
j_setflowlayout	<i>procedure j_setflowlayout (obj , align)</i> Adds a flowlayout manager to dialog obj with the specified alignment .
j_setfocus	<i>function j_setfocus (obj)</i> Directs the input focus to dialog obj .
j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .

j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (obj , row , col)</i> Adds a gridlayout manager to dialog obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (obj , hgap)</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj , top , bottom , left , right)</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj)</i> Removes the current layout manager from dialog obj .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the dialog obj to the specified Position (xpos,ypos).
j_setresizable	<i>procedure j_setresizable (obj , resizable)</i> The dialog cannot be resized, if resizable is J.FALSE .
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes dialog obj to specified width and height .
j_settext	<i>procedure j_settext (obj , str\$)</i> Sets the content or the label of the dialog obj to str .
j_setvgap	<i>procedure j_setvgap (obj , vgap)</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj , color)</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj)</i> Shows the dialog obj .
j_textarea	<i>function j_textarea (obj , rows , columns)</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>function j_textfield (obj , columns)</i> Creates a new textfield component with the specified number of columns and returns its event number.

- j_vscrollbar** *function j_vscrollbar (obj)*
Creates a new vertical scrollbar and returns its event number.
- j_windowlistener** *function j_windowlistener (window , kind)*
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** *function j_focuslistener (obj)*
Adds a new focus listener to focuslistener **obj**, and returns its event number.
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the focuslistener **obj**.
- j_hasfocus** *function j_hasfocus (obj)*
Returns J_TRUE if the focuslistener has the focus, J_FALSE otherwise.

Frame

- j_frame** *function j_frame (label\$)*
Creates a new frame component with the specified **label** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds frame **obj** to container **cont**
- j_alertbox** *procedure j_alertbox (obj , title\$, text\$, button\$)*
Shows a alertbox with the specified **title**, **text** and **button**.
- j_borderpanel** *function j_borderpanel (obj , type)*
Creates a new borderpanel component with the style **type** and returns its event number.
- j_button** *function j_button (obj , label\$)*
Creates a new button component with the specified **label** and returns its event number.
- j_canvas** *function j_canvas (obj , width , height)*
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** *function j_checkbox (obj , label\$)*
Creates a new checkbox component with the specified **label** and returns its event number.
- j_choicebox2** *procedure j_choicebox2 (obj , title\$, text\$, button1\$, button2\$)*
Shows a choicebox with the specified **title**, **text** and two buttons.
- j_choicebox3** *procedure j_choicebox3 (obj , title\$, text\$, button1\$, button2\$, button3\$)*
Shows a choicebox with the specified **title**, **text** and three buttons.
- j_choice** *function j_choice (obj)*
Creates a new choice component and returns its event number.
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *function j_dialog (obj , label\$)*
Creates a new dialog window with the specified **label** and returns its event number.
- j_disable** *procedure j_disable (obj)*
Disables frame **obj** so that it is unresponsive to user interactions

j_dispose	<i>procedure j_dispose (obj)</i> Releases the resources of the frame obj .
j_enable	<i>procedure j_enable (obj)</i> enables the frame obj .
j_filedialog	<i>function\$ j_filedialog (frame , title\$, directory\$)</i> Opens a filedialog box in the specified directory with the specified title and returns the selected filename .
j_fileselect	<i>function\$ j_fileselect (frame , title\$, filter\$)</i> Opens a fileselector box with the preselected filename and the specified title and returns the selected filename .
j_focuslistener	<i>function j_focuslistener (obj)</i> Adds a new focus listener to frame obj , and returns its event number.
j_getfontascent	<i>function j_getfontascent (obj)</i> Returns the ascent (space above the baseline) of the actual font of frame obj .
j_getfontheight	<i>function j_getfontheight (obj)</i> Returns the total pixel height of the actual font of frame obj .
j_getheight	<i>function j_getheight (obj)</i> Returns the height of frame obj .
j_getinheight	<i>function j_getinheight (cont)</i> Returns the height of the client size.
j_getinsets	<i>function j_getinsets (obj , side)</i> Returns the width of the specified inset.
j_getinwidth	<i>function j_getinwidth (cont)</i> Returns the width of the client size.
j_getlayoutid	<i>function j_getlayoutid (obj)</i> Returns the event number of the layoutmanager for containers obj .
j_getlength	<i>function j_getlength (obj)</i> Returns the length of frame 's label or text.
j_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of frame obj .
j_gettext	<i>function\$ j_gettext (obj)</i>

returns the frame's text or label.

- j_getwidth** *function j_getwidth (obj)*
Returns the width of frame **obj**.
- j_getxpos** *function j_getxpos (obj)*
Returns the current horizontal position of frame **obj** in its parent's coordinate space.
- j_getypos** *function j_getypos (obj)*
Returns the current vertical position of frame **obj** in its parent's coordinate space.
- j_graphicbutton** *function j_graphicbutton (obj , filename\$)*
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.
- j_graphiclabel** *function j_graphiclabel (obj , str\$)*
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.
- j_hide** *procedure j_hide (obj)*
Hides the frame **obj**.
- j_hscrollbar** *function j_hscrollbar (obj)*
Creates a new horizontal scrollbar and returns its event number.
- j_isparent** *function j_isparent (obj , cont)*
Returns J_TRUE if **cont** is parent of **obj**, J_FALSE otherwise.
- j_isresizable** *function j_isresizable (obj)*
returns true if frame is resizable, false otherwise
- j_isvisible** *function j_isvisible (obj)*
Returns J_TRUE if **obj** is visible, J_FALSE otherwise.
- j_keylistener** *function j_keylistener (obj)*
Adds a new key listener to frame **obj**, and returns its event number.
- j_label** *function j_label (obj , label\$)*
Creates a new label component with the specified **label** and returns its event number.
- j_led** *function j_led (obj , style , color)*
Creates a new led component with the specified **style** and the specified color **color**.
- j_line** *function j_line (obj , orient , style , length)*
Creates a new line component with the specified **length** and returns its event number.
- j_list** *function j_list (obj , rows)*
Creates a new list component with the specified number of **rows** and returns its event number.

j_menubar	<i>function j_menubar (obj)</i> Creates a new menubar and returns its event number.
j_messagebox	<i>procedure j_messagebox (obj , title\$, text\$)</i> Shows a messagebox with the specified title and text and returns its event number.
j_meter	<i>function j_meter (obj , title\$)</i> Creates a new pointer-instrument with the specified label titel .
j_mouselistener	<i>function j_mouselistener (obj , 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>procedure j_pack (obj)</i> Resizes frame to the minimal size of contained components.
j_panel	<i>function j_panel (obj)</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_printer	<i>function j_printer (frame)</i> Creates a new object, representing a paper of the printer.
j_print	<i>procedure j_print (obj)</i> prints the frame .
j_progressbar	<i>function j_progressbar (obj , orient)</i> Creates a new progressbar with the specified orientation .
j_radiogroup	<i>function j_radiogroup (obj)</i> Creates a new radiogroup and returns its event number.
j_releaseall	<i>procedure j_releaseall (obj)</i> Releases all components from frame obj .
j_release	<i>procedure j_release (obj)</i> Releases frame obj from its parent component (container).
j_scrollpane	<i>function j_scrollpane (obj)</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>procedure j_setalign (obj , align)</i> Sets the alignment in frame obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (obj)</i> Adds a borderlayout manager to frame obj .
j_setborderpos	<i>procedure j_setborderpos (obj , pos)</i>

Moves frame **obj** at a certain position. The outer container needs a border layout manager.

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

j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj)</i> Removes the current layout manager from frame obj .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the frame obj to the specified Position (xpos,ypos).
j_setresizable	<i>procedure j_setresizable (obj , resizable)</i> The frame cannot be resized, if resizable is J_FALSE .
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes frame obj to specified width and height .
j_settext	<i>procedure j_settext (obj , str\$)</i> Sets the content or the label of the frame obj to str .
j_setvgap	<i>procedure j_setvgap (obj , vgap)</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj , color)</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj)</i> Shows the frame obj .
j_textarea	<i>function j_textarea (obj , rows , columns)</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>function j_textfield (obj , columns)</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj)</i> Creates a new vertical scrollbar and returns its event number.
j_windowlistener	<i>function j_windowlistener (window , kind)</i> Adds a new windowlistener to obj , and returns its event number. An event occurs, if the user action is of kind kind .
j_window	<i>function j_window (obj)</i> Creates a new simple window and returns its event number.

Helpmenu

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

Hscrollbar

- j_hscrollbar** *function j_hscrollbar (obj)*
Creates a new horizontal scrollbar and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds hscrollbar **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables hscrollbar **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the hscrollbar **obj**.
- j_enable** *procedure j_enable (obj)*
enables the hscrollbar **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to hscrollbar **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of hscrollbar **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of hscrollbar **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of hscrollbar **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of hscrollbar **obj**.
- j_getvalue** *function j_getvalue (obj)*
Returns the current setting of the scrollbar.

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

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

Graphicbutton

- j_graphicbutton** *function j_graphicbutton (obj , filename\$)*
 Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
 Adds graphicbutton **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
 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** *procedure j_disable (obj)*
 Disables graphicbutton **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
 Releases the resources of the graphicbutton **obj**.
- j_enable** *procedure j_enable (obj)*
 enables the graphicbutton **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
 Adds a new focus listener to graphicbutton **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
 Returns the ascent (space above the baseline) of the actual font of graphicbutton **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
 Returns the total pixel height of the actual font of graphicbutton **obj**.
- j_getheight** *function j_getheight (obj)*
 Returns the height of graphicbutton **obj**.
- j_getparentid** *function j_getparentid (obj)*
 Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
 Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
 Returns the length of **str** of the actual font of graphicbutton **obj**.
- j_getwidth** *function j_getwidth (obj)*
 Returns the width of graphicbutton **obj**.

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

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

Graphiclabel

- j_graphiclabel** *function j_graphiclabel (obj , str\$)*
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds graphiclabel **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables graphiclabel **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the graphiclabel **obj**.
- j_enable** *procedure j_enable (obj)*
enables the graphiclabel **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to graphiclabel **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of graphiclabel **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of graphiclabel **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of graphiclabel **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of graphiclabel **obj**.
- j_getwidth** *function j_getwidth (obj)*
Returns the width of graphiclabel **obj**.

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

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

Image

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

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 *procedure j_drawscaledimage (obj , image , sx , sy , sw , sh , tx , ty , tw , 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 *procedure j_drawstring (obj , x , y , str\$)*

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

j_fillarc *procedure j_fillarc (obj , x , y , rx , ry , arc1 , 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 *procedure j_fillcircle (obj , x , y , r)*

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

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

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

j_fillpolygon *procedure j_fillpolygon (obj , len , x , y)*

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

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

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

j_fillroundrect *procedure j_fillroundrect (obj , x , y , width , height , arcx , 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 *function j_getheight (obj)*

Returns the height of image **obj**.

j_getimage *function j_getimage (obj)*

Copy the contents of image **obj** into an image and return its eventnumber.

j_getimagesource *function j_getimagesource (obj , x , y , w , h , r , g , 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 *function j_getscaledimage (obj , x , y , sw , sh , tw , 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 *function j_getwidth (obj)*

Returns the width of image **obj**.

j_print *procedure j_print (obj)*

prints the image .

j_setxor

procedure j_setxor (obj , 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

procedure j_translate (obj , x , y)

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

KeyListener

- j_keylistener** *function j_keylistener (obj)*
Adds a new key listener to keylistener **obj**, and returns its event number.
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the keylistener **obj**.
- j_getkeychar** *function j_getkeychar (obj)*
Returns the ascii value of the last pressed key.
- j_getkeycode** *function j_getkeycode (obj)*
Returns the integer key code of the last pressed key.

Label

- j_label** *function j_label (obj , label\$)*
Creates a new label component with the specified **label** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds label **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables label **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the label **obj**.
- j_enable** *procedure j_enable (obj)*
enables the label **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to label **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of label **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of label **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of label **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of label **obj**.
- j_gettext** *function\$ j_gettext (obj)*
returns the label 's text or label.
- j_getwidth** *function j_getwidth (obj)*

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

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

Led

- j_led** *function j_led (obj , style , color)*
Creates a new led component with the specified **style** and the specified color **color**.
- j_add** *procedure j_add (obj , cont)*
Adds led **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
Adds a new componentlistener to led **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j_disable** *procedure j_disable (obj)*
Disables led **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the led **obj**.
- j_enable** *procedure j_enable (obj)*
enables the led **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to led **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of led **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of led **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of led **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstate** *function j_getstate (obj)*
Returns **J_TRUE** , if led is selected, **J_FALSE** otherwise.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of led **obj**.
- j_getwidth** *function j_getwidth (obj)*

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

	Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the led obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes led obj to specified width and height .
j_setstate	<i>procedure j_setstate (obj , bool)</i> The led becomes selected, if bool is J.TRUE .
j_show	<i>procedure j_show (obj)</i> Shows the led obj .

List

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

	Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getselect	<i>function j_getselect (obj)</i> Returns the position of currently selected item.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of list obj .
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of list obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of list obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of list obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj)</i> Hides the list obj .
j_insert	<i>function j_insert (obj , pos , label\$)</i> inserts a new item to list obj at position pos with the specified label .
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_iselect	<i>function j_iselect (obj , item)</i> Returns J_TRUE if the particular item is currently selected, J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to list obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj , 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>function j_multiplemode (obj , bool)</i> if bool is J_TRUE , selection mode is turned to multiplemode.
j_popupmenu	<i>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.

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

j_setpos *procedure j_setpos (obj , xpos , ypos)*
Relocates the list **obj** to the specified Position (**xpos,ypos**).

j_setsize *procedure j_setsize (obj , width , height)*
Resizes list **obj** to specified **width** and **height**.

j_show *procedure j_show (obj)*
Shows the list **obj**.

Menu

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

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

j_setshortcut

procedure j_setshortcut (obj , chr)

Changes the shortcut **chr** of the menu .

j_settext

procedure j_settext (obj , str\$)

Sets the content or the label of the menu **obj** to **str**.

Menuitem

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

Meter

- j_meter** *function j_meter (obj , title\$)*
Creates a new pointer-instrument with the specified label **titel**.
- j_add** *procedure j_add (obj , cont)*
Adds meter **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
Adds a new componentlistener to meter **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j_disable** *procedure j_disable (obj)*
Disables meter **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the meter **obj**.
- j_enable** *procedure j_enable (obj)*
enables the meter **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to meter **obj**, and returns its event number.
- j_getdanger** *procedure j_getdanger (obj)*
Returns the danger value of meter **obj**.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of meter **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of meter **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of meter **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of meter **obj**.
- j_getwidth** *function j_getwidth (obj)*

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

	Directs the input focus to meter obj .
j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setmax	<i>function j_setmax (obj , val)</i> Changes the maximum value for the meter to val .
j_setmin	<i>function j_setmin (obj , val)</i> Changes the minimum value for the meter to val .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the meter obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes meter obj to specified width and height .
j_setvalue	<i>procedure j_setvalue (obj , val)</i> Changes the current value of the meter to val .
j_show	<i>procedure j_show (obj)</i> Shows the meter obj .

Mouselistener

- j_mouselistener** *function j_mouselistener (obj , kind)*
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** *procedure j_dispose (obj)*
Releases the resources of the mouselistener **obj**.
- j_getmousebutton** *function j_getmousebutton (mouselistener)*
Returns the latest used mousebutton.
- j_getmousex** *function j_getmousex (mouselistener)*
Returns the current horizontal position of the mouse in its parent's coordinate space.
- j_getmousey** *function j_getmousey (mouselistener)*
Returns the current vertical position of the mouse in its parent's coordinate space.

Panel

- j_panel** *function j_panel (obj)*
Creates a new panel component and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds panel **obj** to container **cont**
- j_borderpanel** *function j_borderpanel (obj , type)*
Creates a new borderpanel component with the style **type** and returns its event number.
- j_button** *function j_button (obj , label\$)*
Creates a new button component with the specified **label** and returns its event number.
- j_canvas** *function j_canvas (obj , width , height)*
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** *function j_checkbox (obj , label\$)*
Creates a new checkbox component with the specified **label** and returns its event number.
- j_choice** *function j_choice (obj)*
Creates a new choice component and returns its event number.
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables panel **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the panel **obj**.
- j_enable** *procedure j_enable (obj)*
enables the panel **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to panel **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of panel **obj**.
- j_getfontheight** *function j_getfontheight (obj)*

	Returns the total pixel height of the actual font of panel obj .
j_getheight	<i>function j_getheight (obj)</i> Returns the height of panel obj .
j_getinheight	<i>function j_getinheight (cont)</i> Returns the height of the client size.
j_getinsets	<i>function j_getinsets (obj , side)</i> Returns the width of the specified inset.
j_getinwidth	<i>function j_getinwidth (cont)</i> Returns the width of the client size.
j_getlayoutid	<i>function j_getlayoutid (obj)</i> Returns the event number of the layoutmanager for containers obj .
j_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of panel obj .
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of panel obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of panel obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of panel obj in its parent's coordinate space.
j_graphicbutton	<i>function j_graphicbutton (obj , filename\$)</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>function j_graphiclabel (obj , str\$)</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>procedure j_hide (obj)</i> Hides the panel obj .
j_hscrollbar	<i>function j_hscrollbar (obj)</i> Creates a new horizontal scrollbar and returns its event number.

j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to panel obj , and returns its event number.
j_label	<i>function j_label (obj , label\$)</i> Creates a new label component with the specified label and returns its event number.
j_led	<i>function j_led (obj , style , color)</i> Creates a new led component with the specified style and the specified color .
j_line	<i>function j_line (obj , orient , style , length)</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>function j_list (obj , rows)</i> Creates a new list component with the specified number of rows and returns its event number.
j_meter	<i>function j_meter (obj , title\$)</i> Creates a new pointer-instrument with the specified label titel .
j_mouselistener	<i>function j_mouselistener (obj , 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>procedure j_pack (obj)</i> Resizes panel to the minimal size of contained components.
j_panel	<i>function j_panel (obj)</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the panel .
j_progressbar	<i>function j_progressbar (obj , orient)</i> Creates a new progressbar with the specified orientation .
j_radiogroup	<i>function j_radiogroup (obj)</i> Creates a new radiogroup and returns its event number.
j_releaseall	<i>procedure j_releaseall (obj)</i> Releases all components from panel obj .

j_release	<i>procedure j_release (obj)</i> Releases panel obj from its parent component (container).
j_scrollpane	<i>function j_scrollpane (obj)</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>procedure j_setalign (obj , align)</i> Sets the alignment in panel obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (obj)</i> Adds a borderlayout manager to panel obj .
j_setborderpos	<i>procedure j_setborderpos (obj , pos)</i> Moves panel obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (obj , r , g , b)</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (obj , r , g , b)</i> Sets the foreground color to the (r , g , b) values.
j_setcursor	<i>function j_setcursor (obj , cursor)</i> Changes the panel 's obj cursor to the specified cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj)</i> Adds a fixlayout manager to panel obj (default layout manager).
j_setflowfill	<i>procedure j_setflowfill (obj , bool)</i> Resizes all containing component to the height (width) of panel obj . Needs a flowlayout manager.
j_setflowlayout	<i>procedure j_setflowlayout (obj , align)</i> Adds a flowlayout manager to panel obj with the specified alignment .
j_setfocus	<i>function j_setfocus (obj)</i> Directs the input focus to panel obj .
j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (obj , row , col)</i> Adds a gridlayout manager to panel obj with the specified rows and columns .

j_sethgap	<i>procedure j_sethgap (obj , hgap)</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj , top , bottom , left , right)</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj)</i> Removes the current layout manager from panel obj .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the panel obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes panel obj to specified width and height .
j_setvgap	<i>procedure j_setvgap (obj , vgap)</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj , color)</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj)</i> Shows the panel obj .
j_textarea	<i>function j_textarea (obj , rows , columns)</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>function j_textfield (obj , columns)</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj)</i> Creates a new vertical scrollbar and returns its event number.

<h2 style="margin: 0;">Popupmenu</h2>

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

j_showpopup*procedure j_showpopup (obj , xpos , ypos)*Shows the popupmenu at specified Position (**xpos,ypos**).

Printer

- j_printer** *function j_printer (frame)*
Creates a new object, representing a paper of the printer.
- j_cliprect** *procedure j_cliprect (obj , x , y , width , height)*
Changes current clipping region to the specified rectangle (**x**, **y**, **width**, **height**).
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the printer **obj**.
- j_drawarc** *procedure j_drawarc (obj , x , y , rx , ry , arc1 , arc2)*
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** *procedure j_drawcircle (obj , x , y , r)*
Draws an unfilled circle with center (**x**, **y**) and radius **x**.
- j_drawimage** *procedure j_drawimage (obj , image , x , y)*
Copies the image, given by its eventnumber **image**, to position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (obj , x , y , w , h , r , g , b)*
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** *procedure j_drawline (obj , x1 , y1 , x2 , y2)*
Draws a line connecting (**x1**,**y1**) and (**x2**,**y2**).
- j_drawoval** *procedure j_drawoval (obj , x , y , rx , ry)*
Draws an unfilled oval with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j_drawpixel** *procedure j_drawpixel (obj , x , y)*
Draws a pixel at (**x**,**y**).
- j_drawpolygon** *procedure j_drawpolygon (obj , len , x , y)*
Draws an unfilled polygon based on first **len** elements in **x** and **y**.
- j_drawpolyline** *procedure j_drawpolyline (obj , len , x , y)*
Draws a series of line segments based on first **len** elements in **x** and **y**.
- j_drawrect** *procedure j_drawrect (obj , x , y , width , height)*
Draws an unfilled rectangle from (**x**,**y**) of size **width** x **height**.
- j_drawroundrect** *procedure j_drawroundrect (obj , x , y , width , height , arcx , arcy)*
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** *procedure j_drawscaledimage (obj , image , sx , sy , sw , sh , tx , ty , tw , 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** *procedure j_drawstring (obj , x , y , str\$)*
Draws text on screen at position (**x**,**y**).
- j_fillarc** *procedure j_fillarc (obj , x , y , rx , ry , arc1 , 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** *procedure j_fillcircle (obj , x , y , r)*
Draws an filled circle with center (**x**, **y**) and radius **x**.
- j_filloval** *procedure j_filloval (obj , x , y , rx , ry)*
Draws an filled oval with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j_fillpolygon** *procedure j_fillpolygon (obj , len , x , y)*
Draws an filled polygon based on first **len** elements in **x** and **y**.
- j_fillrect** *procedure j_fillrect (obj , x , y , width , height)*
Draws an filled rectangle from (**x**,**y**) of size **width** x **height**.
- j_fillroundrect** *procedure j_fillroundrect (obj , x , y , width , height , arcx , 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** *procedure j_print (obj)*
prints the printer .
- j_setxor** *procedure j_setxor (obj , 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** *procedure j_translate (obj , x , y)*
Moves the origin of drawing operations to (**x**, **y**).

Progressbar

- j_progressbar** *function j_progressbar (obj , orient)*
Creates a new progressbar with the specified **orientation**.
- j_add** *procedure j_add (obj , cont)*
Adds progressbar **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
Adds a new componentlistener to progressbar **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j_disable** *procedure j_disable (obj)*
Disables progressbar **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the progressbar **obj**.
- j_enable** *procedure j_enable (obj)*
enables the progressbar **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to progressbar **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of progressbar **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of progressbar **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of progressbar **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of progressbar **obj**.
- j_getwidth** *function j_getwidth (obj)*
Returns the width of progressbar **obj**.

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

j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the progressbar obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes progressbar obj to specified width and height .
j_show	<i>procedure j_show (obj)</i> Shows the progressbar obj .

Radiobutton

- j_radiobutton** *function j_radiobutton (obj , label\$)*
Creates a new radiobutton with the specified **label** and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds radiobutton **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables radiobutton **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the radiobutton **obj**.
- j_enable** *procedure j_enable (obj)*
enables the radiobutton **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to radiobutton **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of radiobutton **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of radiobutton **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of radiobutton **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstate** *function j_getstate (obj)*
Returns **J_TRUE** , if radiobutton is selected, **J_FALSE** otherwise.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of radiobutton **obj**.

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

j_setfocus	<i>function j_setfocus (obj)</i> Directs the input focus to radiobutton obj .
j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the radiobutton obj to the specified Position (xpos,ypos).
j_setradiogroup	<i>function j_setradiogroup (rbutton, , 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>procedure j_setsize (obj , width , height)</i> Resizes radiobutton obj to specified width and height .
j_setstate	<i>procedure j_setstate (obj , bool)</i> The radiobutton becomes selected, if bool is J_TRUE .
j_settext	<i>procedure j_settext (obj , str\$)</i> Sets the content or the label of the radiobutton obj to str .
j_show	<i>procedure j_show (obj)</i> Shows the radiobutton obj .

Sevensegment

- j_sevensegment** *function j_sevensegment (obj , color)*
Creates a new sevensegment display with the specified color **color**.
- j_add** *procedure j_add (obj , cont)*
Adds sevensegment-component **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
Adds a new componentlistener to sevensegment-component **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j_disable** *procedure j_disable (obj)*
Disables sevensegment-component **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the sevensegment-component **obj**.
- j_enable** *procedure j_enable (obj)*
enables the sevensegment-component **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to sevensegment-component **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of sevensegment-component **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of sevensegment-component **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of sevensegment-component **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of sevensegment-component **obj**.
- j_getwidth** *function j_getwidth (obj)*

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

j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the sevensegment–component obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes sevensegment–component obj to specified width and height .
j_setvalue	<i>procedure j_setvalue (obj , val)</i> Changes the current value of the sevensegment–component to val .
j_show	<i>procedure j_show (obj)</i> Shows the sevensegment–component obj .

Scrollpane

- j_scrollpane** *function j_scrollpane (obj)*
Creates a new scrollpane component and returns its event number.
- j_add** *procedure j_add (obj , cont)*
Adds scrollpane **obj** to container **cont**
- j_componentlistener** *function j_componentlistener (obj , kind)*
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** *procedure j_disable (obj)*
Disables scrollpane **obj** so that it is unresponsive to user interactions
- j_dispose** *procedure j_dispose (obj)*
Releases the resources of the scrollpane **obj**.
- j_enable** *procedure j_enable (obj)*
enables the scrollpane **obj**.
- j_focuslistener** *function j_focuslistener (obj)*
Adds a new focus listener to scrollpane **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj)*
Returns the ascent (space above the baseline) of the actual font of scrollpane **obj**.
- j_getfontheight** *function j_getfontheight (obj)*
Returns the total pixel height of the actual font of scrollpane **obj**.
- j_getheight** *function j_getheight (obj)*
Returns the height of scrollpane **obj**.
- j_getparentid** *function j_getparentid (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j_getparent** *function j_getparent (obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j_getstringwidth** *function j_getstringwidth (obj , str\$)*
Returns the length of **str** of the actual font of scrollpane **obj**.
- j_getviewportheight** *function j_getviewportheight (obj)*
Returns the height of the scrollpane 's **obj** port (the area that is shown)

j_getviewportwidth	<i>function j_getviewportwidth (obj)</i> Returns the width of the scrollpane's obj port (the area that is shown)
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of scrollpane obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of scrollpane obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of scrollpane obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj)</i> Hides the scrollpane obj .
j_hscrollbar	<i>function j_hscrollbar (obj)</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to scrollpane obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj , 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>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the scrollpane .
j_release	<i>procedure j_release (obj)</i> Releases scrollpane obj from its parent component (container).
j_setborderpos	<i>procedure j_setborderpos (obj , pos)</i> Moves scrollpane obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (obj , r , g , b)</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (obj , r , g , b)</i> Sets the foreground color to the (r , g , b) values.

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

Textarea

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

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

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

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

Textfield

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

j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getselend	<i>function j_getselend (obj)</i> Returns the ending position of any selected text.
j_getselstart	<i>function j_getselstart (obj)</i> Returns the initial position of any selected text.
j_getseltext	<i>function\$ j_getseltext (obj)</i> Returns the currently selected text of textfield obj .
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of textfield obj .
j_gettext	<i>function\$ j_gettext (obj)</i> returns the textfield 's text or label.
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of textfield obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of textfield obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of textfield obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj)</i> Hides the textfield obj .
j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to textfield obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj , 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>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the textfield .

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

j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the textfield obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes textfield obj to specified width and height .
j_settext	<i>procedure j_settext (obj , str\$)</i> Sets the content or the label of the textfield obj to str .
j_show	<i>procedure j_show (obj)</i> Shows the textfield obj .

Vscrollbar

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

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

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

Window

j_window	<p><i>function j_window (obj)</i> Creates a new simple window and returns its event number.</p>
j_add	<p><i>procedure j_add (obj , cont)</i> Adds window obj to container cont</p>
j_borderpanel	<p><i>function j_borderpanel (obj , type)</i> Creates a new borderpanel component with the style type and returns its event number.</p>
j_button	<p><i>function j_button (obj , label\$)</i> Creates a new button component with the specified label and returns its event number.</p>
j_canvas	<p><i>function j_canvas (obj , width , 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.</p>
j_checkbox	<p><i>function j_checkbox (obj , label\$)</i> Creates a new checkbox component with the specified label and returns its event number.</p>
j_choice	<p><i>function j_choice (obj)</i> Creates a new choice component and returns its event number.</p>
j_componentlistener	<p><i>function j_componentlistener (obj , 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.</p>
j_disable	<p><i>procedure j_disable (obj)</i> Disables window obj so that it is unresponsive to user interactions</p>
j_dispose	<p><i>procedure j_dispose (obj)</i> Releases the resources of the window obj.</p>
j_enable	<p><i>procedure j_enable (obj)</i> enables the window obj.</p>
j_focuslistener	<p><i>function j_focuslistener (obj)</i> Adds a new focus listener to window obj, and returns its event number.</p>
j_getfontascent	<p><i>function j_getfontascent (obj)</i> Returns the ascent (space above the baseline) of the actual font of window obj.</p>
j_getfontheight	<p><i>function j_getfontheight (obj)</i></p>

	Returns the total pixel height of the actual font of window obj .
j_getheight	<i>function j_getheight (obj)</i> Returns the height of window obj .
j_getinheight	<i>function j_getinheight (cont)</i> Returns the height of the client size.
j_getinsets	<i>function j_getinsets (obj , side)</i> Returns the width of the specified inset.
j_getinwidth	<i>function j_getinwidth (cont)</i> Returns the width of the client size.
j_getlayoutid	<i>function j_getlayoutid (obj)</i> Returns the event number of the layoutmanager for containers obj .
j_getparentid	<i>function j_getparentid (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>function j_getparent (obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstringwidth	<i>function j_getstringwidth (obj , str\$)</i> Returns the length of str of the actual font of window obj .
j_getwidth	<i>function j_getwidth (obj)</i> Returns the width of window obj .
j_getxpos	<i>function j_getxpos (obj)</i> Returns the current horizontal position of window obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj)</i> Returns the current vertical position of window obj in its parent's coordinate space.
j_graphicbutton	<i>function j_graphicbutton (obj , filename\$)</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>function j_graphiclabel (obj , str\$)</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>procedure j_hide (obj)</i> Hides the window obj .
j_hscrollbar	<i>function j_hscrollbar (obj)</i> Creates a new horizontal scrollbar and returns its event number.

j_isparent	<i>function j_isparent (obj , cont)</i> Returns J_TRUE if cont is parent of obj , J_FALSE otherwise.
j_isvisible	<i>function j_isvisible (obj)</i> Returns J_TRUE if obj is visible, J_FALSE otherwise.
j_keylistener	<i>function j_keylistener (obj)</i> Adds a new key listener to window obj , and returns its event number.
j_label	<i>function j_label (obj , label\$)</i> Creates a new label component with the specified label and returns its event number.
j_led	<i>function j_led (obj , style , color)</i> Creates a new led component with the specified style and the specified color .
j_line	<i>function j_line (obj , orient , style , length)</i> Creates a new line component with the specified length and returns its event number.
j_list	<i>function j_list (obj , rows)</i> Creates a new list component with the specified number of rows and returns its event number.
j_meter	<i>function j_meter (obj , title\$)</i> Creates a new pointer-instrument with the specified label titel .
j_mouselistener	<i>function j_mouselistener (obj , 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>procedure j_pack (obj)</i> Resizes window to the minimal size of contained components.
j_panel	<i>function j_panel (obj)</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>function j_popupmenu (obj , label\$)</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj)</i> prints the window .
j_progressbar	<i>function j_progressbar (obj , orient)</i> Creates a new progressbar with the specified orientation .
j_radiogroup	<i>function j_radiogroup (obj)</i> Creates a new radiogroup and returns its event number.
j_releaseall	<i>procedure j_releaseall (obj)</i> Releases all components from window obj .

j_release	<i>procedure j_release (obj)</i> Releases window obj from its parent component (container).
j_scrollpane	<i>function j_scrollpane (obj)</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>procedure j_setalign (obj , align)</i> Sets the alignment in window obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (obj)</i> Adds a borderlayout manager to window obj .
j_setborderpos	<i>procedure j_setborderpos (obj , pos)</i> Moves window obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (obj , r , g , b)</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (obj , r , g , b)</i> Sets the foreground color to the (r , g , b) values.
j_setcursor	<i>function j_setcursor (obj , cursor)</i> Changes the window 's obj cursor to the specified cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj)</i> Adds a fixlayout manager to window obj (default layout manager).
j_setflowfill	<i>procedure j_setflowfill (obj , bool)</i> Resizes all containing component to the height (width) of window obj . Needs a flowlayout manager.
j_setflowlayout	<i>procedure j_setflowlayout (obj , align)</i> Adds a flowlayout manager to window obj with the specified alignment .
j_setfocus	<i>function j_setfocus (obj)</i> Directs the input focus to window obj .
j_setfontname	<i>procedure j_setfontname (obj , name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj , name , style , size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj , size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj , style)</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (obj , row , col)</i>

	Adds a gridlayout manager to window obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (obj , hgap)</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj , top , bottom , left , right)</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj , color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj , color)</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj)</i> Removes the current layout manager from window obj .
j_setpos	<i>procedure j_setpos (obj , xpos , ypos)</i> Relocates the window obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj , width , height)</i> Resizes window obj to specified width and height .
j_setvgap	<i>procedure j_setvgap (obj , vgap)</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj , color)</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj)</i> Shows the window obj .
j_textarea	<i>function j_textarea (obj , rows , columns)</i> Creates a new textarea component with the specified number of rows columns and returns its event number.
j_textfield	<i>function j_textfield (obj , columns)</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj)</i> Creates a new vertical scrollbar and returns its event number.
j_windowlistener	<i>function j_windowlistener (window , 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

additem

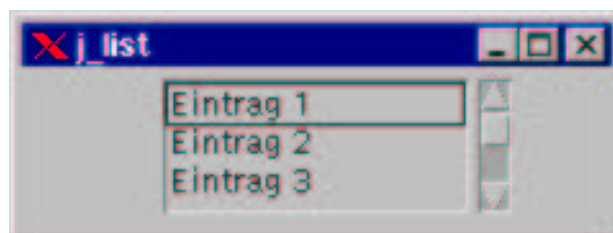
Synopsis subroutine **j_additem** (obj , str\$)

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")  
:
```



add

Synopsis subroutine **j_add** (obj , cont)

Description Adds component **obj** to container **cont**

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

alertbox

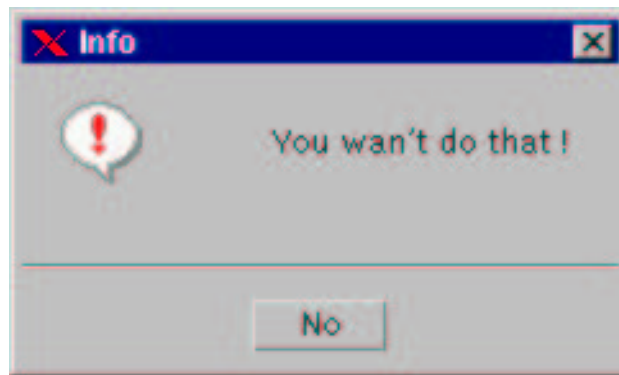
Synopsis subroutine **j_alertbox** (obj , title\$, text\$, button\$)

Description Shows a alertbox with the specified **title**, **text** and **button**. Alertboxes 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 subroutine **j_appendtext** (obj , text\$)

Description Appends the given **text** to the **obj** current text.

Targets Textarea

beep

Synopsis subroutine **j_beep** ()

Description Emits an audio beep.

borderpanel

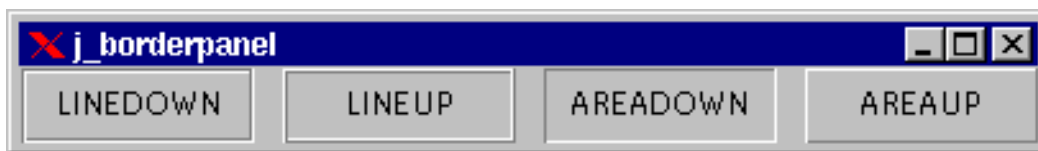
Synopsis function **j_borderpanel** (obj , type)

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	function j_button (obj , label\$)
Description	Creates a new button component with the specified label and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	

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



canvas

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

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 `-1` will be returned.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



checkbox

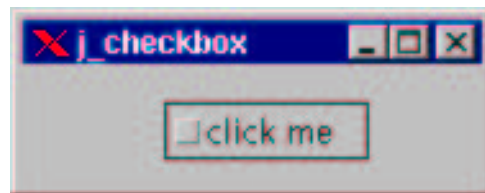
Synopsis function **j_checkbox** (obj , label\$)

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

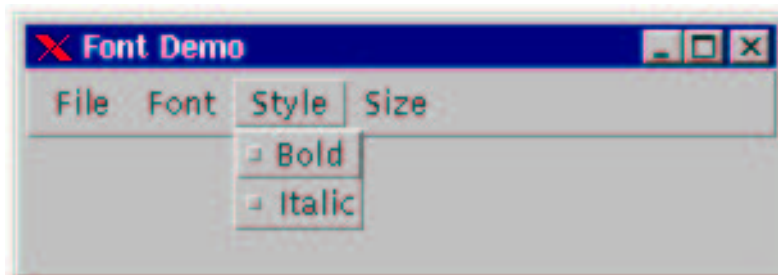
```
:  
frame = j_frame("j_checkbox")  
checkbox = j_checkbox(frame,"click me")  
:
```



checkmenuitem

Synopsis	function j_checkmenuitem (obj , label\$)
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")  
:
```



choicebox2

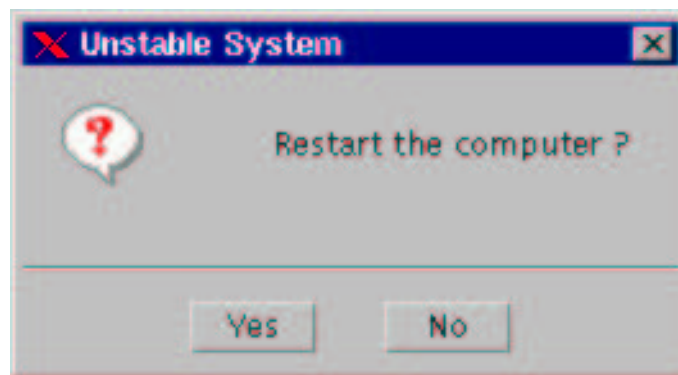
Synopsis subroutine **j_choicebox2** (obj , title\$, text\$, button1\$,
 button2\$)

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")  
:
```



choicebox3

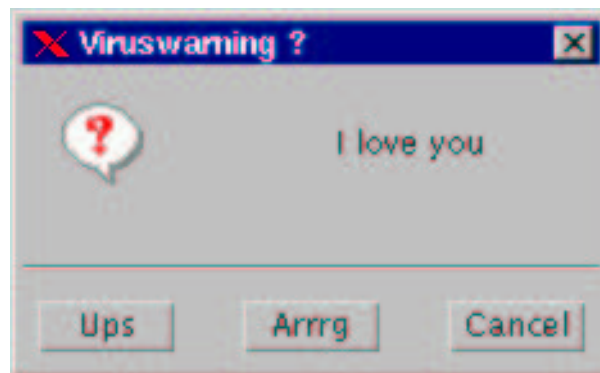
Synopsis subroutine **j_choicebox3** (obj , title\$, text\$, button1\$,
 button2\$, button3\$)

Description Shows a choicebox 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

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



choice

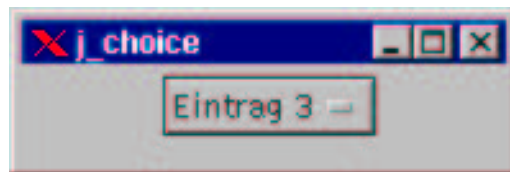
Synopsis function **j_choice** (obj)

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



cliprect

Synopsis	subroutine j_cliprect (obj , x , y , width , height)
Description	Changes current clipping region to the specified rectangle (x , y , width , height).
Targets	Canvas, Image, Printer

componentlistener

Synopsis	function j_componentlistener (obj , kind)
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, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

connect

Synopsis `function j_connect (hostname$)`

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

Example

```
:  
if( j_connect("atan.japi.de") = J_FALSE ) then  
  
    or  
  
if( j_connect("127.0.0.1") = J_FALSE ) then  
:
```


delete

Synopsis	subroutine j_delete (obj , start , end)
Description	Deletes text from starting position start to ending position end .
Targets	Textarea

deselect

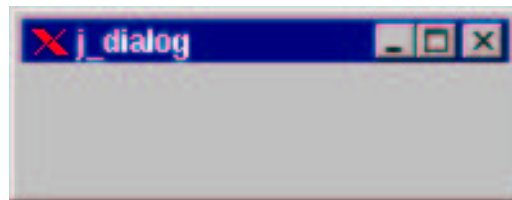
Synopsis	function j_deselect (obj , item)
Description	Deselects the item at the designated position item , if selected.
Targets	List

dialog

Synopsis	function j_dialog (obj , label\$)
Description	Creates a new dialog window with the specified label and returns its event number.
Targets	Frame

Example

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



disable

Synopsis	subroutine j_disable (obj)
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, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu

dispose

Synopsis	subroutine j_dispose (obj)
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, Led, Progressbar, Meter, Sevensegment, Canvas, Image, Printer, Keylistener, Focuslistener, Mouselistener

drawarc

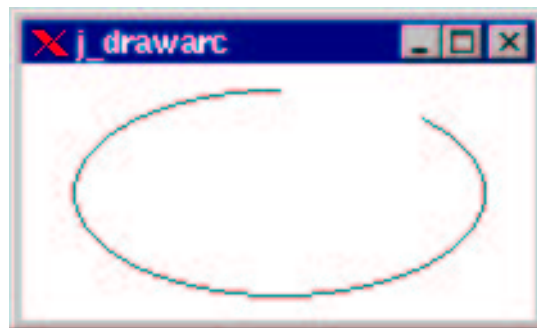
Synopsis subroutine **j_drawarc** (obj , x , y , rx , ry , arc1 , arc2)

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

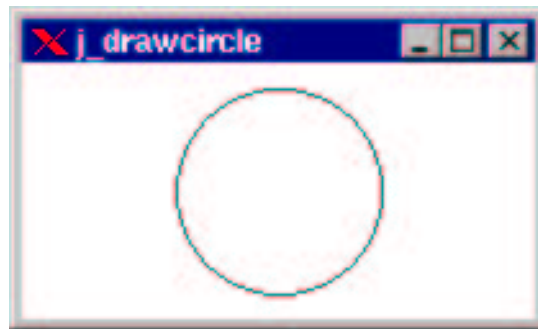
Synopsis subroutine **j_drawcircle** (obj , x , y , r)

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)  
:
```



drawimagesource

Synopsis	subroutine j_drawimagesource (obj , x , y , w , h , r , g , b)
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

drawimage

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

drawline

Synopsis subroutine **j_drawline** (obj , x1 , y1 , x2 , y2)

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)  
:
```

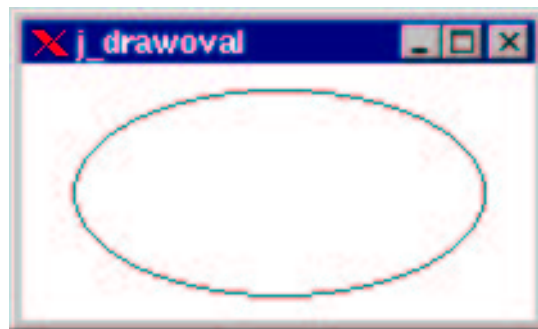


drawoval

- Synopsis subroutine **j_drawoval** (obj , x , y , rx , ry)
- 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

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



drawpixel

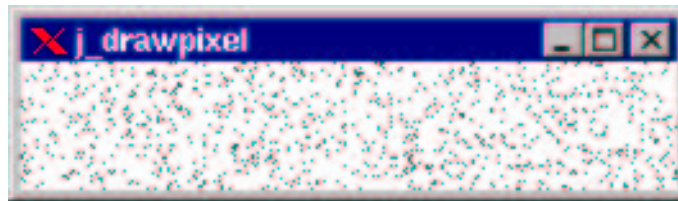
Synopsis subroutine **j_drawpixel** (obj , x , y)

Description Draws a pixel at (x,y).

Targets Canvas, Image, Printer

Example

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



drawpolygon

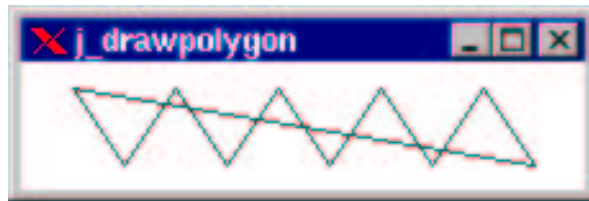
Synopsis subroutine **j_drawpolygon** (obj , len , x , y)

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

Targets Canvas, Image, Printer

Example

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



drawpolyline

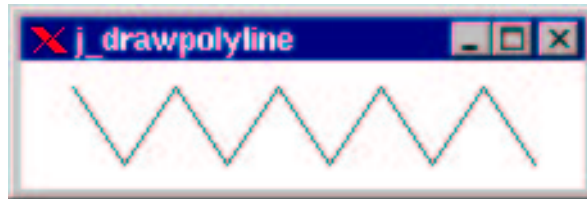
Synopsis subroutine **j_drawpolyline** (obj , len , x , y)

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

Targets Canvas, Image, Printer

Example

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



drawrect

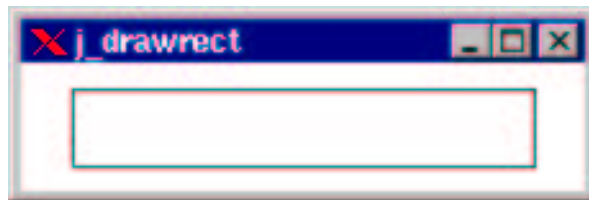
Synopsis subroutine **j_drawrect** (obj , x , y , width , height)

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

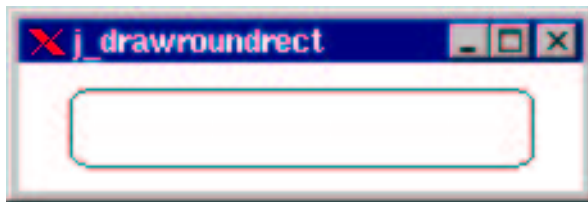
Synopsis subroutine **j_drawroundrect** (obj , x , y , width , height ,
 arcx , arcy)

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

Targets Canvas, Image, Printer

Example

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



drawscaledimage

Synopsis	subroutine j_drawscaledimage (obj , image , sx , sy , sw , sh , tx , ty , tw , th)
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

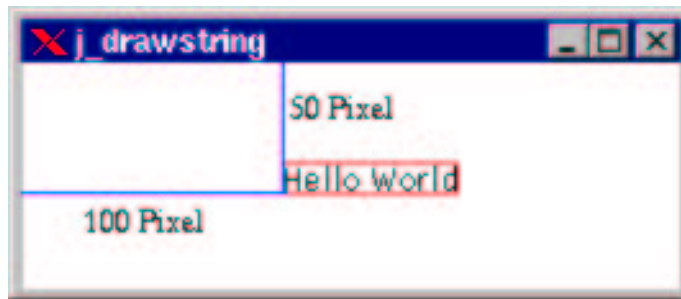
Synopsis subroutine **j_drawstring** (obj , x , y , str\$)

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

Targets Canvas, Image, Printer

Example

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



enable

Synopsis	subroutine j_enable (obj)
Description	enables the component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

filedialog

Synopsis `function$ j_filedialog (frame , title$, directory$)`

Description Opens a filedialog box in the specified **directory** with the specified **title** and returns the selected **filename**. If **title** contains `"/S"` the `SAVE-filedialog` will be called. The substring `"/S"` will be removed.

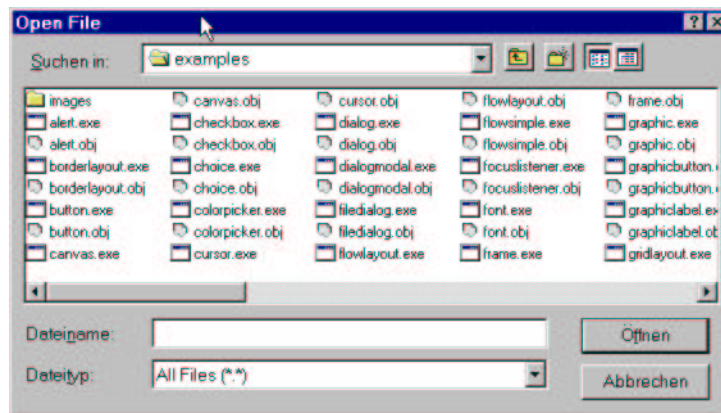
Targets `Frame`

Example

```

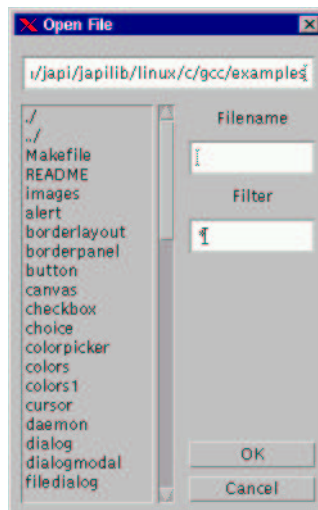
:
filename = j_filedialog(frame,"Save/S File","..")
:

```



fileselect

Synopsis	<code>function\$ j_fileselect (frame , title\$, filter\$)</code>
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","*") : </pre>



fillarc

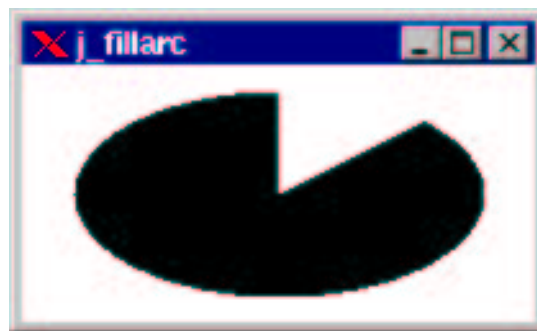
Synopsis subroutine **j_fillarc** (obj , x , y , rx , ry , arc1 , arc2)

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 subroutine **j_fillcircle** (obj , x , y , r)

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

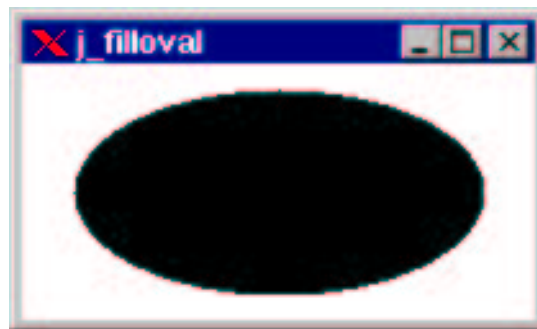
Synopsis subroutine **j_filloval** (obj , x , y , rx , ry)

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 subroutine **j_fillpolygon** (obj , len , x , y)

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

Targets Canvas, Image, Printer

Example

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



fillrect

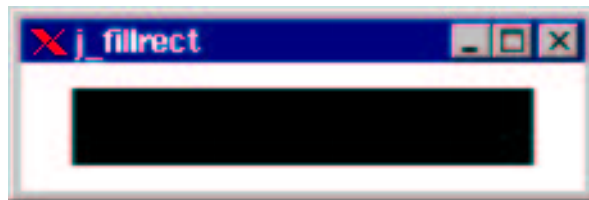
Synopsis subroutine **j_fillrect** (obj , x , y , width , height)

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

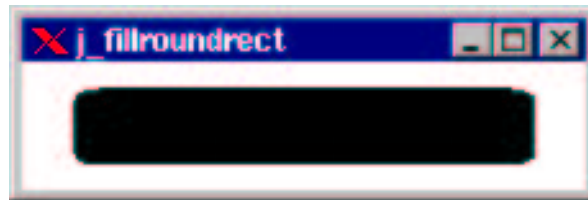
Synopsis subroutine **j_fillroundrect** (obj , x , y , width , height ,
 arcx , arcy)

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,j,5)  
:
```



focuslistener

Synopsis	function j_focuslistener (obj)
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, Led, Progressbar, Meter, Sevensegment

frame

Synopsis function **j_frame** (label\$)

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

Example

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



getaction

Synopsis function **j_getaction** ()

Description returns the next event, or 0 if no event available

getcolumns

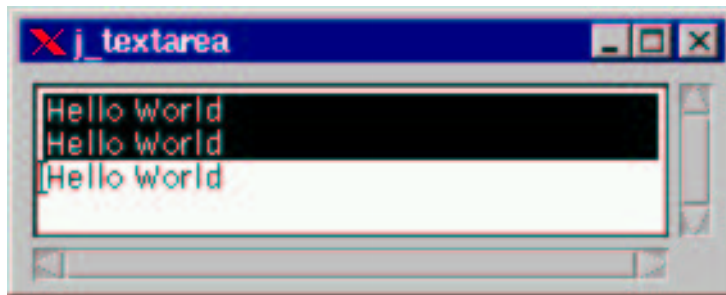
Synopsis subroutine **j_getcolumns** (obj)

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 function **j_getcurpos** (obj)

Description Returns the position, in characters, of the text cursor.

Targets Textarea, Textfield

getdanger

Synopsis	subroutine j_getdanger (obj)
Description	Returns the danger value of component obj .
Targets	Meter

getfontascent

Synopsis	function j_getfontascent (obj)
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, Led, Progressbar, Meter, Sensesegment

getfontheight

Synopsis	function j_getfontheight (obj)
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, Led, Progressbar, Meter, Sevensegment

getheight

Synopsis function **j_getheight** (obj)

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, Led, Progressbar,
Meter, Sevensegment, Image

Example

```
:  
label = j_getlabel(frame,"Hello World")  
print j_getheight(label)  
:
```

```
> 22
```

getimagesource

Synopsis function **j_getimagesource** (obj , x , y , w , h , r , g , b)

Description 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**

Targets Canvas, Image

getimage

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

getinheight

Synopsis function **j_getinheight** (cont)

Description Returns the height of the client size.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
frame = j_frame("Hello World")
j_setsize(frame,300,400)
print j_getinheight(frame)
:
> 370
```

getinsets

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

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")
print j_getinsets(frame,J_TOP) ,j_getinsets(frame,J_BOTTOM),
      j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT)
:
> 25 5 5 6

```



getinwidth

Synopsis function **j_getinwidth** (cont)

Description Returns the width of the client size.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
frame = j_frame("Hello World")
j_setsize(frame,300,400)
print j_getinwidth(frame)
:
> 289
```

getitemcount

Synopsis	function j_getitemcount (obj)
Description	Returns the number of items of component obj .
Targets	List, Choice

getitem

Synopsis function\$ **j_getitem** (obj , item)

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

Targets List, Choice

getkeychar

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

getkeycode

Synopsis	function j_getkeycode (obj)
Description	Returns the integer key code of the last pressed key.
Targets	KeyListener

getlayoutid

Synopsis	function j_getlayoutid (obj)
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	function j_getlength (obj)
Description	Returns the length of component 's label or text.
Targets	Textarea, Textfield, Dialog, Frame, Button, MenuItem, CheckBox-MenuItem, Menu, HelpMenu, Popupmenu

getmousebutton

Synopsis function **j_getmousebutton** (mouset listener)

Description Returns the latest used mousebutton. The return value is:

- J_LEFT left mousebutton
- J_CENTER middle mousebutton
- J_RIGHT right mousebutton

Targets Mousetlistener

getmousex

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

getmousey

Synopsis function **j_getmousey** (mouset listener)

Description Returns the current vertical position of the mouse in its parent's coordinate space.

Targets Mousetlistener

getparentid

Synopsis `function j_getparentid (obj)`

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, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Example

```

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

```



getparent

Synopsis function **j_getparent** (obj)

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, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Example

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



getrows

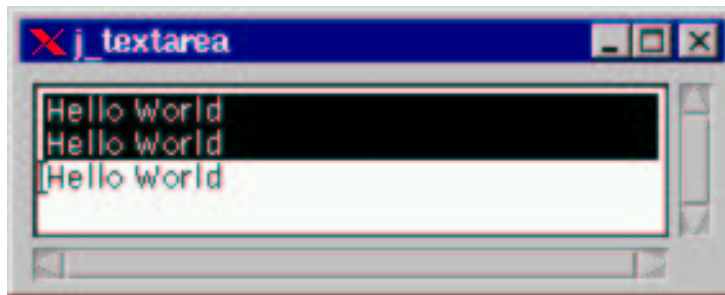
Synopsis subroutine **j_getrows** (obj)

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

Targets Textarea, Gridlayout

Example

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



getscaledimage

Synopsis	function j_getscaledimage (obj , x , y , sw , sh , tw , th)
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 function **j_getscreenheight** ()

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

Example

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

getscreenwidth

Synopsis function **j_getscreenwidth** ()

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

Example

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


getselect

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

getselend

Synopsis	function j_getselend (obj)
Description	Returns the ending position of any selected text.
Targets	Textarea, Textfield

getselstart

Synopsis	function j_getselstart (obj)
Description	Returns the initial position of any selected text.
Targets	Textarea, Textfield

getseltext

Synopsis function\$ **j_getseltext** (obj)

Description Returns the currently selected text of component **obj**.

Targets Textarea, Textfield

getstate

Synopsis	function j_getstate (obj)
Description	Returns J_TRUE , if component is selected, J_FALSE otherwise.
Targets	Checkbox, Radiobutton, CheckmenuItem, Led

getstringwidth

Synopsis	function j_getstringwidth (obj , str\$)
Description	Returns the length of str of the actual font of component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

gettext

Synopsis	function\$ j_gettext (obj)
Description	returns the component 's text or label.
Targets	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield
Example	<pre>: label = j_label(frame,"Hello World") print j_gettext(label) : > Hello World</pre>

getvalue

Synopsis function **j_getvalue** (obj)

Description Returns the current setting of the scrollbar.

Targets Scrollbar

getviewportheight

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

getviewportwidth

Synopsis	function j_getviewportwidth (obj)
Description	Returns the width of the component 's obj port (the area that is shown)
Targets	Scrollpane

getwidth

Synopsis	function j_getwidth (obj)
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, Led, Progressbar, Meter, Sevensegment, Image
Example	<pre>: label = j_getlabel(frame,"Hello World") print j_getwidth(label) : > 84</pre>

getxpos

Synopsis	function j_getxpos (obj)
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, Led, Progressbar, Meter, Sensegment

getypos

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

graphicbutton

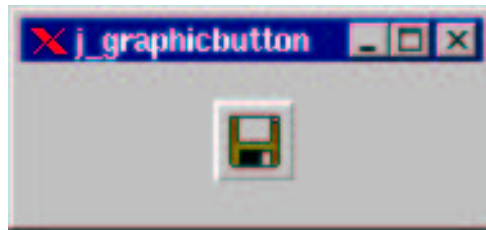
Synopsis `function j_graphicbutton (obj , filename$)`

Description Creates a new `graphicbutton` component with the image loaded from **filename** and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



graphiclabel

Synopsis `function j_graphiclabel (obj , str$)`

Description Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



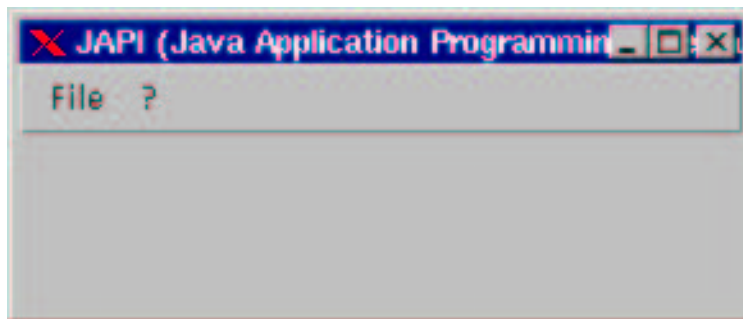
hasfocus

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

helpmenu

Synopsis	function j_helpmenu (obj , label\$)
Description	Creates a new helpmenu component with the specified label and returns its event number.
Targets	Menubar
Example	

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



hide

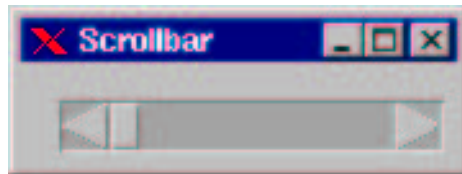
Synopsis subroutine **j_hide** (obj)

Description Hides the component **obj**.

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

hscrollbar

- Synopsis `function j_hscrollbar (obj)`
- 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      function **j\_image** ( width , height )

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    | function <b>j_insert</b> ( obj , pos , label\$ )                                                    |
| Description | inserts a new item to component <b>obj</b> at position <b>pos</b> with the specified <b>label</b> . |
| Targets     | List, Choice                                                                                        |

|            |
|------------|
| inserttext |
|------------|

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_inserttext</b> ( obj , text\$ , pos )                          |
| Description | Places additional text within the component at the given position <b>pos</b> . |
| Targets     | Textarea                                                                       |

|          |
|----------|
| isparent |
|----------|

|             |                                                                                                                                                                                                                                                                                                     |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_isparent</b> ( obj , cont )                                                                                                                                                                                                                                                           |
| Description | Returns J_TRUE if <b>cont</b> is parent of <b>obj</b> , J_FALSE otherwise.                                                                                                                                                                                                                          |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup |

|             |
|-------------|
| isresizable |
|-------------|

|             |                                                         |
|-------------|---------------------------------------------------------|
| Synopsis    | function <b>j_isresizable</b> ( obj )                   |
| Description | returns true if component is resizable, false otherwise |
| Targets     | Dialog, Frame                                           |



|          |
|----------|
| isselect |
|----------|

|             |                                                                                        |
|-------------|----------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_isselect</b> ( obj , item )                                              |
| Description | Returns J_TRUE if the particular <b>item</b> is currently selected, J_FALSE otherwise. |
| Targets     | List                                                                                   |

isvisible

|             |                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_isvisible</b> ( obj )                                                                                                                                                                                    |
| Description | Returns J_TRUE if <b>obj</b> is visible, J_FALSE otherwise.                                                                                                                                                            |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

|             |
|-------------|
| keylistener |
|-------------|

|             |                                                                                                                                                                                                                         |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_keylistener</b> ( obj )                                                                                                                                                                                   |
| Description | Adds a new key listener to component <b>obj</b> , and returns its event number.                                                                                                                                         |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment |

|       |
|-------|
| label |
|-------|

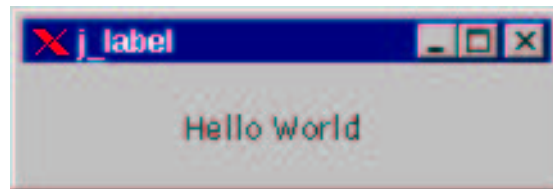
Synopsis      function **j\_label** ( obj , label\$ )

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

Targets      Panel, Borderpanel, Window, Dialog, Frame

Example

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



## led

Synopsis            function **j\_led** ( obj , style , color )

Description        Creates a new led component and returns its event number. The LEDs shape could be round if **style=J\_ROUND** or a rectangle if **style=J\_RECT**. The color could be one of the predefined colors (eg. J\_RED, J\_GREEN).

Targets            Panel, Borderpanel, Window, Dialog, Frame

Example

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



## line

Synopsis           function **j\_line** ( obj , orient , style , length )

Description       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 `-1` will be returned. The parameter **orient** specifies the orientation of the line:

- `J_HORIZONTAL` : horizontal line
- `J_VERTICAL` : vertical line

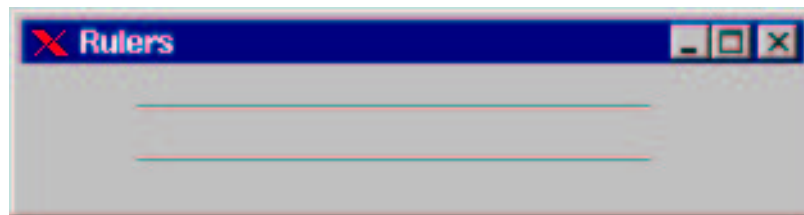
The Parameter **style** specifies the linestyle:

- `J_LINEDOWN` : etched-in linestyle.
- `J_LINEUP` : etched-out linestyle.

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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

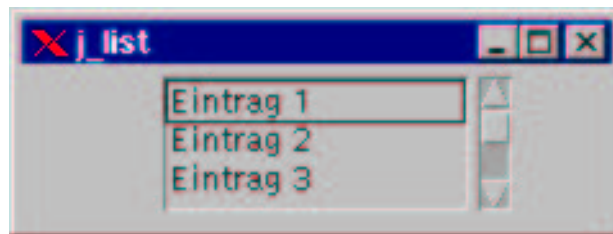


## list

|             |                                                                                                     |
|-------------|-----------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_list</b> ( obj , rows )                                                               |
| Description | Creates a new list component with the specified number of <b>rows</b> and returns its event number. |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                                                           |

### Example

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



loadimage

Synopsis           function **j\_loadimage** ( filename\$ )

Description       Loads the Image from file **filename** and returns its eventnumber.  
The file could be of the following format:

- GIF
- JPEG
- BMP
- PPM

Example

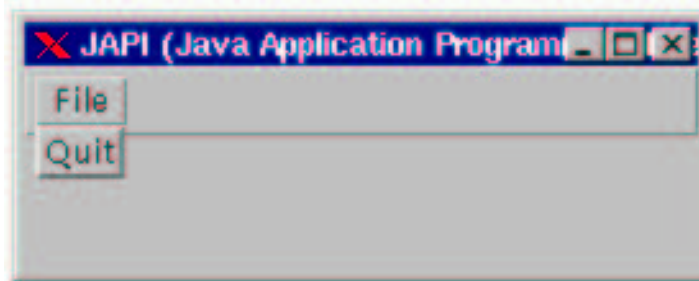
```
:
image = j_loadimage("mandel.jpg")
:
```



## menubar

|             |                                                     |
|-------------|-----------------------------------------------------|
| Synopsis    | function <b>j_menubar</b> ( obj )                   |
| 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

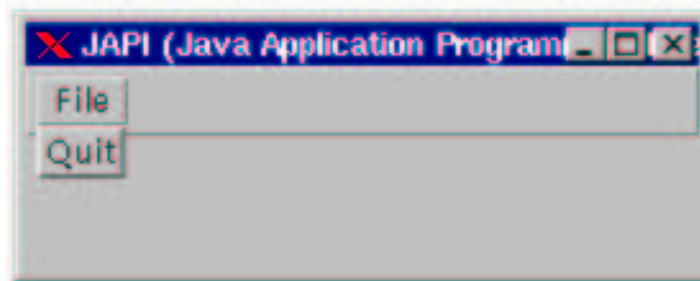
Synopsis      `function j_menuitem ( obj , label$ )`

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")
:
```



## menu

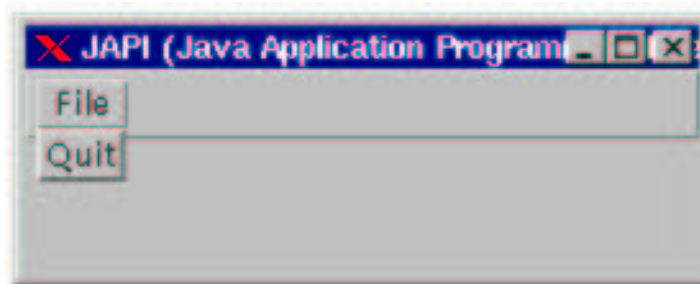
Synopsis      `function j_menu ( obj , str$ )`

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

Targets      Menubar, Menu

Example

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



messagebox

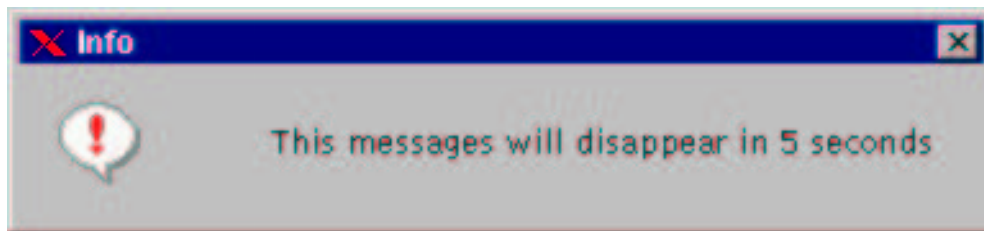
Synopsis            subroutine **j\_messagebox**    ( obj , title\$ , text\$ )

Description       Shows a messagebox with the specified **title** and **text** and returns its event number. In the case of error **-1** will be returned. A Messagebox generates an event, if the close icon is clicked.

Targets            Frame

Example

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



## meter

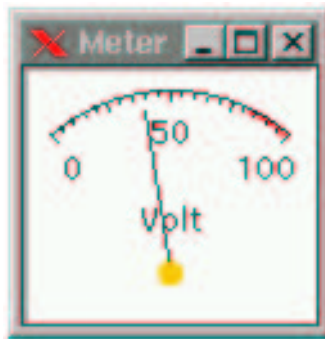
Synopsis            function **j\_meter** ( obj , title\$ )

Description        Creates a new pointer-instrument with the specified label **titel** and returns its event number. The meter has predifined values from 0 to 100. This can be canged via `j_setmin()` and `j_setmax()`. A danger value is set to 80 and can be justified with `j_setdanger()`.

Targets            Panel, Borderpanel, Window, Dialog, Frame

Example

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



mouselistener

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

|              |
|--------------|
| multiplemode |
|--------------|

Synopsis           function **j\_multiplemode** ( obj , bool )

Description       if **bool** is J\_TRUE , selection mode is turned to multiplemode.

Targets           List

|            |
|------------|
| nextaction |
|------------|

Synopsis      function **j\_nextaction** ( )

Description      Waits for the next event.



## pack

Synopsis            subroutine **j\_pack** ( obj )

Description       Resizes component to the minimal size of contained components.

Targets            Panel, Borderpanel, Window, Dialog, Frame

### Example

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



## panel

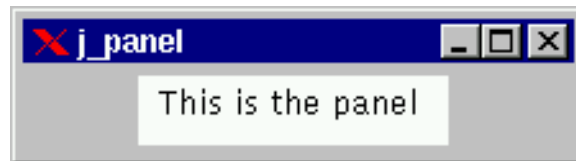
Synopsis           function **j\_panel** ( obj )

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

Targets           Panel, Borderpanel, Window, Dialog, Frame

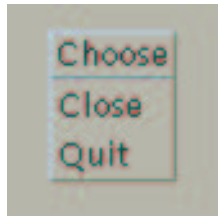
Example

```
:
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    | function <b>j_popupmenu</b> ( obj , label\$ )                                                                                                                                                                          |
| Description | Creates a new popupmenu with the specified <b>label</b> and returns its event number.                                                                                                                                  |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |
| Example     | <pre>: choose = j_popupmenu(frame,"Choose") close  = j_menuitem(choose,"Close") quit   = j_menuitem(choose,"Quit") j_showpopup(choose,100,100) :</pre>                                                                 |



|         |
|---------|
| printer |
|---------|

|             |                                                                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_printer</b> ( frame )                                                                                                                                                                                                                                                          |
| 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>:<br/>printer = j_printer(frame)<br/>j_drawimage(printer,image,100,100)<br/>:</pre>                                                                                                                                                                                                     |

## print

- Synopsis**            subroutine **j\_print** ( obj )
- Description**        prints the component . With X-Windows all components have Motif-look.  
If component is a printer, the actual page will be closed, and a new page will be opened. The pages are not jet printed. To print all pages call `j_dispose(printer)`;
- Targets**            Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Canvas, Image, Printer

**Example**

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



## progressbar

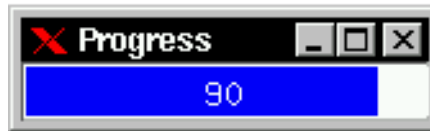
Synopsis            `function j_progressbar ( obj , orient )`

Description        Creates a new progressbar with the specified **orientation** and returns its event number. Orientation could be `J_HORIZONTAL` or `J_VERTICAL`. The progressbar has predefined values from 0 to 100. This can be changed via `j_setmin()` and `j_setmax()`.

Targets            Panel, Borderpanel, Window, Dialog, Frame

Example

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



|      |
|------|
| quit |
|------|

Synopsis            subroutine **j\_quit** ( )

Description        Cancels the connection to the JAPI Kernel.

## radiobutton

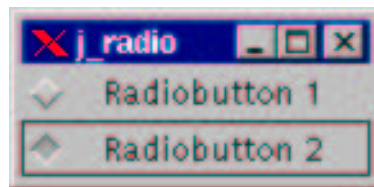
Synopsis            `function j_radiobutton ( obj , label$ )`

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

Targets            Radiogroup

Example

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





## radiogroup

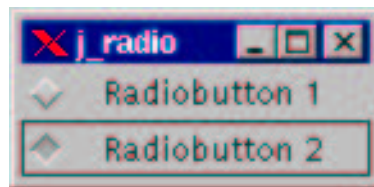
Synopsis            `function j_radiogroup ( obj )`

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      function **j\_random** ( )

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

|            |
|------------|
| releaseall |
|------------|

|             |                                                     |
|-------------|-----------------------------------------------------|
| Synopsis    | subroutine <b>j_releaseall</b> ( obj )              |
| Description | Releases all components from component <b>obj</b> . |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame           |

|         |
|---------|
| release |
|---------|

|             |                                                                                                                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_release</b> ( obj )                                                                                                                                                                                   |
| Description | Releases component <b>obj</b> from its parent component (container).                                                                                                                                                  |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

|           |
|-----------|
| removeall |
|-----------|

|             |                                        |
|-------------|----------------------------------------|
| Synopsis    | function <b>j_removeall</b> ( obj )    |
| Description | Removes all items from the component . |
| Targets     | List, Choice                           |

|            |
|------------|
| removeitem |
|------------|

Synopsis      function **j\_removeitem** ( obj , item\$ )

Description      remove the first occurrence of **item** from the component .

Targets      List, Choice

|        |
|--------|
| remove |
|--------|

Synopsis      function **j\_remove** ( obj , item )

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

Targets      List, Choice

|             |
|-------------|
| replacetext |
|-------------|

|             |                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_replacetext</b> ( obj , text\$ , start , end )                                                   |
| Description | Replaces the text from starting position <b>start</b> to ending position <b>end</b> with the given <b>text</b> . |
| Targets     | Textarea                                                                                                         |



saveimage

|             |                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_saveimage</b> ( obj , filename\$ , filetype )                                                                                                                                        |
| Description | Saves the components image to file <b>filename</b> . The specified file format can be: <ul style="list-style-type: none"><li>• J_BMP Win32 Bitmap Format</li><li>• J_PPM Portable pixmap</li></ul> |
| Example     | <pre>: if(j_saveimage(canvas,"mandel.bmp",J_BMP) = J_FALSE)     print "Error saving Bitmap file" :</pre>                                                                                           |

## scrollpane

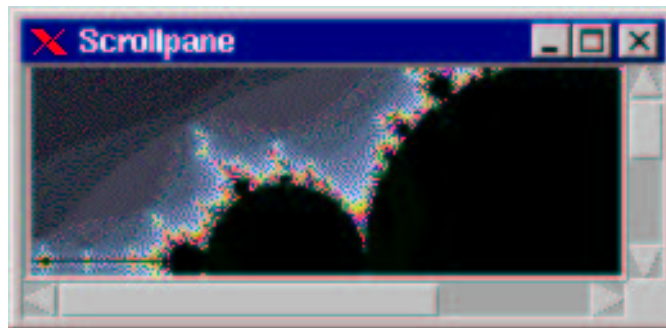
Synopsis           function **j\_scrollpane** ( obj )

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)
:
```



|           |
|-----------|
| selectall |
|-----------|

|             |                                         |
|-------------|-----------------------------------------|
| Synopsis    | subroutine <b>j_selectall</b> ( obj )   |
| Description | Selects all the text in the component . |
| Targets     | Textarea, Textfield                     |

select

Synopsis      function **j\_select** ( obj , item )

Description      Makes the given **item** the selected one for the component .

Targets      List, Choice

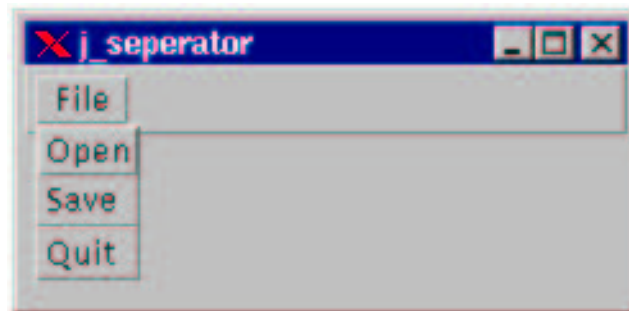
|            |
|------------|
| selecttext |
|------------|

|             |                                                                                  |
|-------------|----------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_selecttext</b> ( obj , start , end )                             |
| Description | Selects text from starting position <b>start</b> to ending position <b>end</b> . |
| Targets     | Textarea, Textfield                                                              |

## seperator

- Synopsis            subroutine **j\_seperator** ( obj )
- 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    | subroutine <b>j_setalign</b> ( obj , align )                                             |
| Description | Sets the alignment in component <b>obj</b> to <b>align</b> . Needs a flowlayout Manager. |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                                                |

|             |
|-------------|
| setblockinc |
|-------------|

Synopsis      function **j\_setblockinc** ( obj , val )

Description      Changes the block increment amount for the component to **val**.

Targets      Scrollbar



|                 |
|-----------------|
| setborderlayout |
|-----------------|

|             |                                                       |
|-------------|-------------------------------------------------------|
| Synopsis    | subroutine <b>j_setborderlayout</b> ( obj )           |
| Description | Adds a borderlayout manager to component <b>obj</b> . |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame             |

|              |
|--------------|
| setborderpos |
|--------------|


|             |                                                                                                                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setborderpos</b> ( obj , pos )                                                                                                                                                                        |
| Description | Moves component <b>obj</b> at a certain position. The outer container needs a border layout manager.                                                                                                                  |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

## setcolorbg

|             |                                                                                                                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setcolorbg</b> ( obj , r , g , , b )                                                                                                                                                                  |
| Description | Sets the background color to the (r, g, b) values.                                                                                                                                                                    |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

### Example

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



Hello World

setcolor


Synopsis            subroutine **j\_setcolor**    ( obj , r , g , , b )

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, Led, Progressbar, Meter, Sevensegment

Example

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



## setcolumns

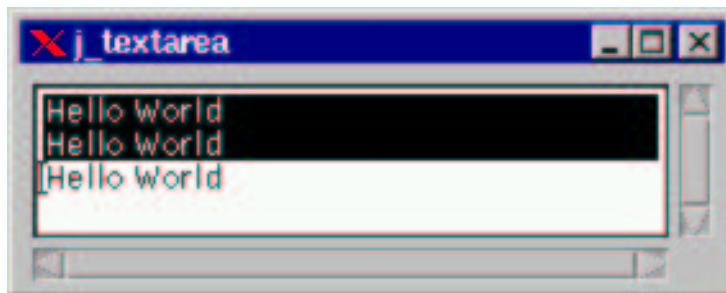
Synopsis            subroutine **j\_setcolumns**    ( obj , columns )

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    | subroutine <b>j_setcurpos</b> ( obj , pos )                                   |
| Description | Change the location of the text cursor to the specified position <b>pos</b> . |
| Targets     | Textarea, Textfield                                                           |

|           |
|-----------|
| setcursor |
|-----------|

|             |                                                                                                                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_setcursor</b> ( obj , cursor )                                                                                                                                                                          |
| Description | Changes the component 's <b>obj</b> cursor to the specified <b>cursor</b> .                                                                                                                                           |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

|           |
|-----------|
| setdanger |
|-----------|

Synopsis            subroutine **j\_setdanger** ( obj , val )

Description        Changes the danger value of component **obj** to **val**.

Targets            Meter



|          |
|----------|
| setdebug |
|----------|

Synopsis            subroutine **j\_setdebug** ( level )

Description        Sets the debuglevel to **level**.

|             |
|-------------|
| setechochar |
|-------------|

|             |                                                                                         |
|-------------|-----------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setechochar</b> ( obj , chr )                                           |
| Description | Changes the character <b>chr</b> that is used to echo all user input in the component . |
| Targets     | Textfield                                                                               |

|             |
|-------------|
| seteditable |
|-------------|

|             |                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_seteditable</b> ( obj , bool )                                                     |
| Description | Allows to make the component editable ( <b>bool=J.TRUE</b> ) or read-only ( <b>bool=J.FALSE</b> ). |
| Targets     | Textarea, Textfield                                                                                |

|              |
|--------------|
| setfixlayout |
|--------------|

|             |                                                                            |
|-------------|----------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setfixlayout</b> ( obj )                                   |
| Description | Adds a fixlayout manager to component <b>obj</b> (default layout manager). |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                                  |

|             |
|-------------|
| setflowfill |
|-------------|

|             |                                                                                                              |
|-------------|--------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setflowfill</b> ( obj , bool )                                                               |
| Description | Resizes all containing component to the height (width) of component <b>obj</b> . Needs a flowlayout manager. |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                                                                    |

|               |
|---------------|
| setflowlayout |
|---------------|

|             |                                                                                         |
|-------------|-----------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setflowlayout</b> ( obj , align )                                       |
| Description | Adds a flowlayout manager to component <b>obj</b> with the specified <b>alignment</b> . |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                                               |

|          |
|----------|
| setfocus |
|----------|

|             |                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_setfocus</b> ( obj )                                                                                                                                                                                     |
| Description | Directs the input focus to component <b>obj</b> .                                                                                                                                                                      |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

setfontname

Synopsis            subroutine **j\_setfontname**    ( obj , name )

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, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

Example

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





## setfontsize

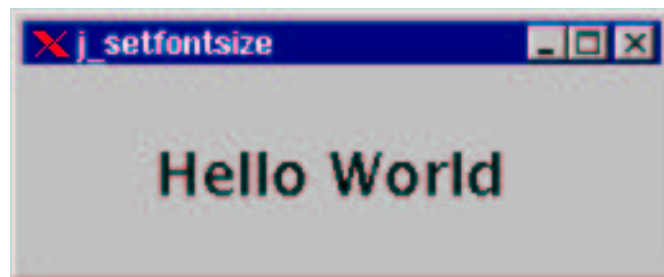
Synopsis            subroutine **j\_setfontsize**    ( obj , size )

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

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

Example

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



## setfontstyle

Synopsis            subroutine **j\_setfontstyle** ( obj , style )

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, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu

Example

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



## setfont

|             |                                                                                                                                                                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setfont</b> ( obj , name , style , size )                                                                                                                                                                                                                    |
| Description | Changes the font to the given characteristics <b>name</b> , <b>style</b> and <b>size</b> .                                                                                                                                                                                   |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popumenu |

### Example

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



|               |
|---------------|
| setgridlayout |
|---------------|

|             |                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setgridlayout</b> ( obj , row , col )                                                 |
| Description | Adds a gridlayout manager to component <b>obj</b> with the specified <b>rows</b> and <b>columns</b> . |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                                                             |

## sethgap

Synopsis            subroutine **j\_sethgap**    ( obj , hgap )

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            subroutine **j\_seticon**    ( frame , icon )

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

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

## setimage

Synopsis            subroutine **j\_setimage** ( obj , image )

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            subroutine **j\_setinsets**    ( obj , top , bottom , left , right )

Description        Set the insets to the specified values.

Targets            Panel, Borderpanel, Window, Dialog, Frame

Example

```

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

> 25 5 5 6

```





|        |
|--------|
| setmax |
|--------|

|             |                                                             |
|-------------|-------------------------------------------------------------|
| Synopsis    | function <b>j_setmax</b> ( obj , val )                      |
| Description | Changes the maximum value for the component to <b>val</b> . |
| Targets     | Scrollbar, Meter, Progress                                  |

|        |
|--------|
| setmin |
|--------|

Synopsis      function **j\_setmin** ( obj , val )

Description      Changes the minimum value for the component to **val**.

Targets      Scrollbar, Meter, Progress

|                 |
|-----------------|
| setnamedcolorbg |
|-----------------|

|             |                                                                                                                                                                                                                         |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setnamedcolorbg</b> ( obj , color )                                                                                                                                                                     |
| Description | Sets the background color to a predefined <b>color</b> .                                                                                                                                                                |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

|               |
|---------------|
| setnamedcolor |
|---------------|

|             |                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setnamedcolor</b> ( obj , color )                                                                                                                                                                      |
| Description | Sets the foreground color to a predefined <b>color</b> .                                                                                                                                                               |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

|             |
|-------------|
| setnolayout |
|-------------|

|             |                                                                |
|-------------|----------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setnolayout</b> ( obj )                        |
| Description | Removes the current layout manager from component <b>obj</b> . |
| Targets     | Panel, Borderpanel, Window, Dialog, Frame                      |

|       |
|-------|
| start |
|-------|

Synopsis            subroutine **j\_start**    ( port )

Description        Replace the default Port by **port**. This can be usefull if the default port is used by an other application, or if you want to start several independent kernels on one machine. This functions must be called before calling j\_start();

Example

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

|        |
|--------|
| setpos |
|--------|

|             |                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setpos</b> ( obj , xpos , ypos )                                                                                                                                                                       |
| Description | Relocates the component <b>obj</b> to the specified Position ( <b>xpos,ypos</b> ).                                                                                                                                     |
| Targets     | Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment |

|               |
|---------------|
| setradiogroup |
|---------------|

|             |                                                                                                                                                                                      |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_setradiogroup</b> ( rbutton, , rgroup )                                                                                                                                |
| Description | Sets radiobuttons <b>rbutton</b> group to be the specified radiogroup <b>rgroup</b> . If the radiobuttons is already in a different radiogroup, it is first taken out of that group. |
| Targets     | Radiobutton                                                                                                                                                                          |



|              |
|--------------|
| setresizable |
|--------------|

Synopsis            subroutine **j\_setresizable**    ( obj , resizable )

Description        The component cannot be resized, if **resizable** is J\_FALSE .

Targets            Dialog, Frame

Example

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

setrows

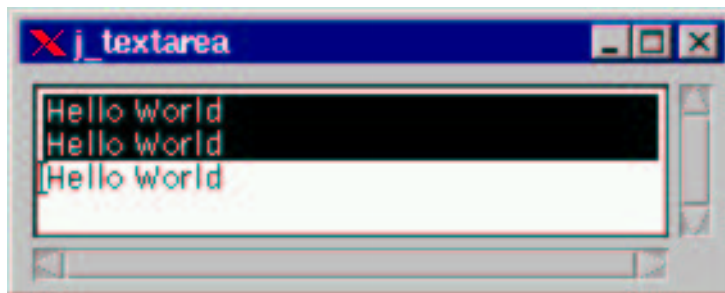
Synopsis            subroutine **j\_setrows**    ( obj , rows )

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    | subroutine <b>j_setshortcut</b> ( obj , chr )         |
| Description | Changes the shortcut <b>chr</b> of the component .    |
| Targets     | MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu |

setsize

Synopsis            subroutine **j\_setsize** ( obj , width , height )

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, Led, Progressbar, Meter, Sevensegment

Example

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



|              |
|--------------|
| setslidesize |
|--------------|

Synopsis      function **j\_setslidesize** ( obj , val )

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

Targets      Scrollbar

|          |
|----------|
| setstate |
|----------|

Synopsis            subroutine **j\_setstate** ( obj , bool )

Description        The component becomes selected, if **bool** is J\_TRUE .

Targets            Checkbox, Radiobutton, Checkmenuitem, Led

## settext


Synopsis            subroutine **j\_settext** ( obj , str\$ )

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

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



Goodbye

|            |
|------------|
| setunitinc |
|------------|

Synopsis      function **j\_setunitinc** ( obj , val )

Description      Changes the unit increment amount for the component to **val**

Targets      Scrollbar



|          |
|----------|
| setvalue |
|----------|

|             |                                                            |
|-------------|------------------------------------------------------------|
| Synopsis    | subroutine <b>j_setvalue</b> ( obj , val )                 |
| Description | Changes the current value of the component to <b>val</b> . |
| Targets     | Scrollbar, Progress, Meter, Sevensegment                   |

setvgap

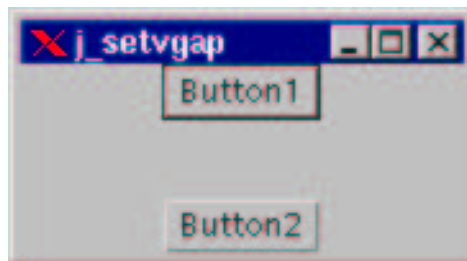
Synopsis            subroutine **j\_setvgap**    ( obj , vgap )

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

Targets            Panel, Borderpanel, Window, Dialog, Frame

Example

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



|        |
|--------|
| setxor |
|--------|

Synopsis            subroutine **j\_setxor** ( obj , bool )

Description        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.

Targets            Canvas, Image, Printer

## sevensegment

Synopsis           function **j\_sevensegment** ( obj , color )

Description       Creates a new sevensegment display and returns its event number. The color could be one of the predefined colors (eg. J\_RED, J\_GREEN).

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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



|           |
|-----------|
| showpopup |
|-----------|

|             |                                                                 |
|-------------|-----------------------------------------------------------------|
| Synopsis    | subroutine <b>j_showpopup</b> ( obj , xpos , ypos )             |
| Description | Shows the component at specified Position ( <b>xpos,ypos</b> ). |
| Targets     | Popupmenu                                                       |

|      |
|------|
| show |
|------|

Synopsis            subroutine **j\_show** ( obj )

Description       Shows the component **obj**.

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

|       |
|-------|
| sleep |
|-------|

Synopsis           function **j\_sleep** ( msec )

Description       Suspends the execution for **msec** milliseconds.

|       |
|-------|
| start |
|-------|

Synopsis      `function j_start ()`

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

Example

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



|      |
|------|
| sync |
|------|

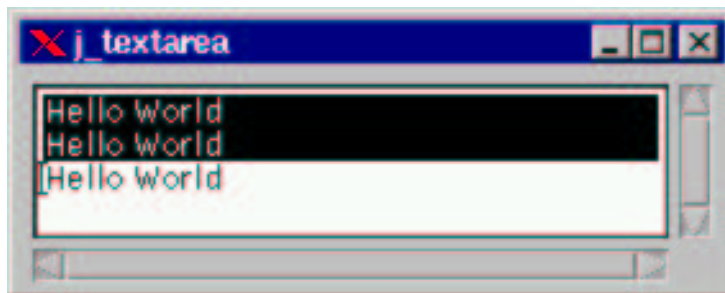
Synopsis            subroutine **j\_sync** ( )

Description        Synchronizes the application with the JAPI kernel.

## textarea

|             |                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------|
| Synopsis    | function <b>j_textarea</b> ( obj , rows , columns )                                                             |
| Description | Creates a new textarea component with the specified number of <b>rows columns</b> 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      function **j\_textfield** ( obj , columns )

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

Targets      Panel, Borderpanel, Window, Dialog, Frame

Example

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



|           |
|-----------|
| translate |
|-----------|

Synopsis      subroutine **j\_translate** ( obj , x , y )

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

Targets      Canvas, Image, Printer

## vscrollbar

- Synopsis            `function j_vscrollbar ( obj )`
- Description        `Creates a new vertical scrollbar and returns its event number.`
- Targets            `Panel, Borderpanel, Window, Dialog, Frame, Scrollpane`
- Example
- ```
:  
scroll=j_vscrollbar(frame)  
j_setpos(scroll,120,40)  
j_setsize(scroll,20,100)  
:
```



windowlistener

Synopsis `function j_windowlistener (window , kind)`

Description 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**:

- **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

window

Synopsis function **j_window** (obj)

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