

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	81
	Radiobutton	84
	Sevensegment	87
	ScrollPane	90
	Textarea	93
	Textfield	97
	Vscrollbar	101
	Window	104

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

	Returns the height of button obj .
j_getlength	<i>function j_getlength (obj: integer) return integer;</i> Returns the length of button 's label or text.
j_getparentid	<i>function j_getparentid (obj: integer) return integer;</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: integer) return integer;</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: integer; str: string) return integer;</i> Returns the length of str of the actual font of button obj .
j_gettext	<i>function j_gettext (obj: integer; str: string) return string;</i> returns the button 's text or label.
j_getwidth	<i>function j_getwidth (obj: integer) return integer;</i> Returns the width of button obj .
j_getxpos	<i>function j_getxpos (obj: integer) return integer;</i> Returns the current horizontal position of button obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj: integer) return integer;</i> Returns the current vertical position of button obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj: integer);</i> Hides the button obj .
j_isparent	<i>function j_isparent (obj: integer; cont: integer) return integer;</i> Returns true if cont is parent of obj , false otherwise.
j_isvisible	<i>function j_isvisible (obj: integer) return integer;</i> Returns true if obj is visible, false otherwise.
j_keylistener	<i>function j_keylistener (obj: integer) return integer;</i> Adds a new key listener to button obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) return integer;</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: integer; label: string) return integer;</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj: integer);</i> prints the button .

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

Borderpanel

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

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

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

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

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

Canvas

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

	Draws an unfilled oval with the center (\mathbf{x}, \mathbf{y}) and the horizontal radius \mathbf{rx} and the vertical radius \mathbf{ry} .
j_drawpixel	<i>procedure j_drawpixel (obj: integer; x: integer; y: integer) ;</i> Draws a pixel at (\mathbf{x}, \mathbf{y}) .
j_drawpolygon	<i>procedure j_drawpolygon (obj: integer; len: integer; x: integer_array; y: integer_array) ;</i> Draws an unfilled polygon based on first len elements in \mathbf{x} and \mathbf{y} .
j_drawpolyline	<i>procedure j_drawpolyline (obj: integer; len: integer; x: integer_array; y: integer_array) ;</i> Draws a series of line segments based on first len elements in \mathbf{x} and \mathbf{y} .
j_drawrect	<i>procedure j_drawrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;</i> Draws an unfilled rectangle from (\mathbf{x}, \mathbf{y}) of size width x height .
j_drawroundrect	<i>procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;</i> Draws an unfilled rectangle from (\mathbf{x}, \mathbf{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: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;</i> Copy the contents of the rectangular area defined by \mathbf{x}, \mathbf{y} , width sw , and height sh of the image to position $(\mathbf{tx}, \mathbf{ty})$. The area will be scaled to target width th and target height th .
j_drawstring	<i>procedure j_drawstring (obj: integer; x: integer; y: integer; str: string) ;</i> Draws text on screen at position (\mathbf{x}, \mathbf{y}) .
j_enable	<i>procedure j_enable (obj: integer) ;</i> enables the canvas obj .
j_fillarc	<i>procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;</i> Draws an filled arc from angle arc1 to angle arc2 with the center (\mathbf{x}, \mathbf{y}) and the horizontal radius rx and the vertical radius ry .
j_fillcircle	<i>procedure j_fillcircle (obj: integer; x: integer; y: integer; r: integer) ;</i> Draws an filled circle with center (\mathbf{x}, \mathbf{y}) and radius x .
j_filloval	<i>procedure j_filloval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;</i> Draws an filled oval with the center (\mathbf{x}, \mathbf{y}) and the horizontal radius rx and the vertical radius ry .
j_fillpolygon	<i>procedure j_fillpolygon (obj: integer; len: integer; x: integer_array; y: integer_array) ;</i> Draws an filled polygon based on first len elements in \mathbf{x} and \mathbf{y} .

- j_fillrect** *procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
 Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j_fillroundrect** *procedure j_fillroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
 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** *function j_focuslistener (obj: integer) return integer;*
 Adds a new focus listener to canvas **obj**, and returns its event number.
- j_getfontascent** *function j_getfontascent (obj: integer) return integer;*
 Returns the ascent (space above the baseline) of the actual font of canvas **obj**.
- j_getfontheight** *function j_getfontheight (obj: integer) return integer;*
 Returns the total pixel height of the actual font of canvas **obj**.
- j_getheight** *function j_getheight (obj: integer) return integer;*
 Returns the height of canvas **obj**.
- j_getimage** *function j_getimage (obj: integer) return integer;*
 Copy the contents of canvas **obj** into an image and return its eventnumber.
- j_getimagesource** *function j_getimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: integer_array; g: integer_array; b: integer_array) return integer;*
 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** *function j_getparentid (obj: integer) return integer;*
 Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getparent** *function j_getparent (obj: integer) return integer;*
 Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j_getscaledimage** *function j_getscaledimage (obj: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer) return integer;*
 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** *function j_getstringwidth (obj: integer; str: string) return integer;*
 Returns the length of **str** of the actual font of canvas **obj**.
- j_getwidth** *function j_getwidth (obj: integer) return integer;*
 Returns the width of canvas **obj**.
- j_getxpos** *function j_getxpos (obj: integer) return integer;*
 Returns the current horizontal position of canvas **obj** in its parent's coordinate space.

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

j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Changes the font to the given style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Relocates the canvas obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Resizes canvas obj to specified width and height .
j_setxor	<i>procedure j_setxor (obj: integer; bool: boolean) ;</i> Changes painting mode to XOR mode, if bool = true . In this mode, drawing the same object in the same color at the same location twice has no net effect.
j_show	<i>procedure j_show (obj: integer) ;</i> Shows the canvas obj .
j_translate	<i>procedure j_translate (obj: integer; x: integer; y: integer) ;</i> Moves the origin of drawing operations to (x, y).

Checkbox

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

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

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

Checkmenuitem

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

Choice

j_choice	<p><i>function j_choice (obj: integer) return integer;</i> Creates a new choice component and returns its event number.</p>
j_additem	<p><i>procedure j_additem (obj: integer; str: string) ;</i> adds a new item containing str to choice obj.</p>
j_add	<p><i>procedure j_add (obj: integer; cont: integer) ;</i> Adds choice obj to container cont</p>
j_componentlistener	<p><i>function j_componentlistener (obj: integer; kind: integer) return integer;</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: integer) ;</i> Disables choice obj so that it is unresponsive to user interactions</p>
j_dispose	<p><i>procedure j_dispose (obj: integer) ;</i> Releases the resources of the choice obj.</p>
j_enable	<p><i>procedure j_enable (obj: integer) ;</i> enables the choice obj.</p>
j_focuslistener	<p><i>function j_focuslistener (obj: integer) return integer;</i> Adds a new focus listener to choice obj, and returns its event number.</p>
j_getfontascent	<p><i>function j_getfontascent (obj: integer) return integer;</i> Returns the ascent (space above the baseline) of the actual font of choice obj.</p>
j_getfontheight	<p><i>function j_getfontheight (obj: integer) return integer;</i> Returns the total pixel height of the actual font of choice obj.</p>
j_getheight	<p><i>function j_getheight (obj: integer) return integer;</i> Returns the height of choice obj.</p>
j_getitemcount	<p><i>function j_getitemcount (obj: integer) return integer;</i> Returns the number of items of choice obj.</p>
j_getitem	<p><i>function j_getitem (obj: integer; item: integer; str: string) return string;</i> returns the label of the given item.</p>
j_getparentid	<p><i>function j_getparentid (obj: integer) return integer;</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: integer) return integer;</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: integer) return integer;</i> Returns the position of currently selected item.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: string) return integer;</i> Returns the length of str of the actual font of choice obj .
j_getwidth	<i>function j_getwidth (obj: integer) return integer;</i> Returns the width of choice obj .
j_getxpos	<i>function j_getxpos (obj: integer) return integer;</i> Returns the current horizontal position of choice obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj: integer) return integer;</i> Returns the current vertical position of choice obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Hides the choice obj .
j_insert	<i>function j_insert (obj: integer; pos: integer; label: string) return integer;</i> inserts a new item to choice obj at position pos with the specified label .
j_isparent	<i>function j_isparent (obj: integer; cont: integer) return integer;</i> Returns true if cont is parent of obj , false otherwise.
j_isvisible	<i>function j_isvisible (obj: integer) return integer;</i> Returns true if obj is visible, false otherwise.
j_keylistener	<i>function j_keylistener (obj: integer) return integer;</i> Adds a new key listener to choice obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) return integer;</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: integer; label: string) return integer;</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj: integer) ;</i> prints the choice .
j_release	<i>procedure j_release (obj: integer) ;</i> Releases choice obj from its parent component (container).
j_removeall	<i>function j_removeall (obj: integer) return integer;</i> Removes all items from the choice .
j_removeitem	<i>function j_removeitem (obj: integer; item: string) return integer;</i>

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

Dialog

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

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

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

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

j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Adds a gridlayout manager to dialog obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Removes the current layout manager from dialog obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Relocates the dialog obj to the specified Position (xpos,ypos).
j_setresizable	<i>procedure j_setresizable (obj: integer; resizable: boolean) ;</i> The dialog cannot be resized, if resizable is false .
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Resizes dialog obj to specified width and height .
j_settext	<i>procedure j_settext (obj: integer; str: string) ;</i> Sets the content or the label of the dialog obj to str .
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) return integer;</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj: integer) ;</i> Shows the dialog obj .
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) return integer;</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: integer; columns: integer) return integer;</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) return integer;</i> Creates a new vertical scrollbar and returns its event number.
j_windowlistener	<i>function j_windowlistener (window: integer; kind: integer) return integer;</i> Adds a new windowlistener to obj , and returns its event number. An event occurs, if the user action is of kind kind .

Focuslistener

- j_focuslistener** *function j_focuslistener (obj: integer) return integer;*
Adds a new focus listener to focuslistener **obj**, and returns its event number.
- j_dispose** *procedure j_dispose (obj: integer) ;*
Releases the resources of the focuslistener **obj**.
- j_hasfocus** *function j_hasfocus (obj: integer) return integer;*
Returns true if the focuslistener has the focus, false otherwise.

Frame

- j_frame** *function j_frame (label: string) return integer;*
Creates a new frame component with the specified **label** and returns its event number.
- j_add** *procedure j_add (obj: integer; cont: integer);*
Adds frame **obj** to container **cont**
- j_alertbox** *procedure j_alertbox (obj: integer; title: string; text: string; button: string);*
Shows a alertbox with the specified **title**, **text** and **button**.
- j_borderpanel** *function j_borderpanel (obj: integer; type: integer) return integer;*
Creates a new borderpanel component with the style **type** and returns its event number.
- j_button** *function j_button (obj: integer; label: string) return integer;*
Creates a new button component with the specified **label** and returns its event number.
- j_canvas** *function j_canvas (obj: integer; width: integer; height: integer) return integer;*
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: integer; label: string) return integer;*
Creates a new checkbox component with the specified **label** and returns its event number.
- j_choicebox2** *procedure j_choicebox2 (obj: integer; title: string; text: string; button1: string; button2: string);*
Shows a choicebox with the specified **title**, **text** and two buttons.
- j_choicebox3** *procedure j_choicebox3 (obj: integer; title: string; text: string; button1: string; button2: string; button3: string);*
Shows a choicebox with the specified **title**, **text** and three buttons.
- j_choice** *function j_choice (obj: integer) return integer;*
Creates a new choice component and returns its event number.
- j_componentlistener** *function j_componentlistener (obj: integer; kind: integer) return integer;*
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: integer; label: string) return integer;*
Creates a new dialog window with the specified **label** and returns its event number.
- j_disable** *procedure j_disable (obj: integer);*

Disables frame **obj** so that it is unresponsive to user interactions

j_dispose	<i>procedure j_dispose (obj: integer) ;</i> Releases the resources of the frame obj .
j_enable	<i>procedure j_enable (obj: integer) ;</i> enables the frame obj .
j_filedialog	<i>function j_filedialog (frame: integer; title: string; directory: string; filename: string) return string;</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: integer; title: string; filter: string; filename: string) return string;</i> Opens a fileslector box with the preselected filename and the specified title and returns the selected filename .
j_focuslistener	<i>function j_focuslistener (obj: integer) return integer;</i> Adds a new focus listener to frame obj , and returns its event number.
j_getfontascent	<i>function j_getfontascent (obj: integer) return integer;</i> Returns the ascent (space above the baseline) of the actual font of frame obj .
j_getfontheight	<i>function j_getfontheight (obj: integer) return integer;</i> Returns the total pixel height of the actual font of frame obj .
j_getheight	<i>function j_getheight (obj: integer) return integer;</i> Returns the height of frame obj .
j_getinheight	<i>function j_getinheight (cont: integer) return integer;</i> Returns the height of the client size.
j_getinsets	<i>function j_getinsets (obj: integer; side: integer) return integer;</i> Returns the width of the specified inset.
j_getinwidth	<i>function j_getinwidth (cont: integer) return integer;</i> Returns the width of the client size.
j_getlayoutid	<i>function j_getlayoutid (obj: integer) return integer;</i> Returns the event number of the layoutmanager for containers obj .
j_getlength	<i>function j_getlength (obj: integer) return integer;</i> Returns the length of frame 's label or text.
j_getparentid	<i>function j_getparentid (obj: integer) return integer;</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: integer) return integer;</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: integer; str: string) return integer;</i> Returns the length of str of the actual font of frame obj .
j_gettext	<i>function j_gettext (obj: integer; str: string) return string;</i> returns the frame 's text or label.
j_getwidth	<i>function j_getwidth (obj: integer) return integer;</i> Returns the width of frame obj .
j_getxpos	<i>function j_getxpos (obj: integer) return integer;</i> Returns the current horizontal position of frame obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj: integer) return integer;</i> Returns the current vertical position of frame obj in its parent's coordinate space.
j_graphicbutton	<i>function j_graphicbutton (obj: integer; filename: string) return integer;</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>function j_graphiclabel (obj: integer; str: string) return integer;</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>procedure j_hide (obj: integer);</i> Hides the frame obj .
j_hscrollbar	<i>function j_hscrollbar (obj: integer) return integer;</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>function j_isparent (obj: integer; cont: integer) return integer;</i> Returns true if cont is parent of obj , false otherwise.
j_isresizable	<i>function j_isresizable (obj: integer) return boolean;</i> returns true if frame is resizable, false otherwise
j_isvisible	<i>function j_isvisible (obj: integer) return integer;</i> Returns true if obj is visible, false otherwise.
j_keylistener	<i>function j_keylistener (obj: integer) return integer;</i> Adds a new key listener to frame obj , and returns its event number.
j_label	<i>function j_label (obj: integer; label: string) return integer;</i> Creates a new label component with the specified label and returns its event number.
j_led	<i>function j_led (obj: integer; style: integer; color: integer) return integer;</i> Creates a new led component with the specified style and the specified color .
j_line	<i>function j_line (obj: integer; orient: integer; style: integer; length: integer) return integer;</i>

	Creates a new line component with the specified length and returns its event number.
j_list	<i>function j_list (obj: integer; rows: integer) return integer;</i> Creates a new list component with the specified number of rows and returns its event number.
j_menubar	<i>function j_menubar (obj: integer) return integer;</i> Creates a new menubar and returns its event number.
j_messagebox	<i>procedure j_messagebox (obj: integer; title: string; text: string) ;</i> Shows a messagebox with the specified title and text and returns its event number.
j_meter	<i>function j_meter (obj: integer; title: string) return integer;</i> Creates a new pointer-instrument with the specified label titel .
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) return integer;</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: integer) ;</i> Resizes frame to the minimal size of contained components.
j_panel	<i>function j_panel (obj: integer) return integer;</i> Creates a new panel component and returns its event number.
j_popupmenu	<i>function j_popupmenu (obj: integer; label: string) return integer;</i> Creates a new popupmenu with the specified label and returns its event number.
j_printer	<i>function j_printer (frame: integer) return integer;</i> Creates a new object, representing a paper of the printer.
j_print	<i>procedure j_print (obj: integer) ;</i> prints the frame .
j_progressbar	<i>function j_progressbar (obj: integer; orient: integer) return integer;</i> Creates a new progressbar with the specified orientation .
j_radiogroup	<i>function j_radiogroup (obj: integer) return integer;</i> Creates a new radiogroup and returns its event number.
j_releaseall	<i>procedure j_releaseall (obj: integer) ;</i> Releases all components from frame obj .
j_release	<i>procedure j_release (obj: integer) ;</i> Releases frame obj from its parent component (container).
j_scrollpane	<i>function j_scrollpane (obj: integer) return integer;</i> Creates a new scrollpane component and returns its event number.
j_setalign	<i>procedure j_setalign (obj: integer; align: integer) ;</i>

	Sets the alignment in frame obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (obj: integer) ;</i> Adds a borderlayout manager to frame obj .
j_setborderpos	<i>procedure j_setborderpos (obj: integer; pos: integer) ;</i> Moves frame obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (obj: integer; r: integer; g,: integer; b: integer) ;</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (obj: integer; r: integer; g,: integer; b: integer) ;</i> Sets the foreground color to the (r , g , b) values.
j_setcursor	<i>function j_setcursor (obj: integer; cursor: integer) return integer;</i> Changes the frame 's obj cursor to the specified cursor .
j_setfixlayout	<i>procedure j_setfixlayout (obj: integer) ;</i> Adds a fixlayout manager to frame obj (default layout manager).
j_setflowfill	<i>procedure j_setflowfill (obj: integer; bool: boolean) ;</i> Resizes all containing component to the height (width) of frame obj . Needs a flowlayout manager.
j_setflowlayout	<i>procedure j_setflowlayout (obj: integer; align: integer) ;</i> Adds a flowlayout manager to frame obj with the specified alignment .
j_setfocus	<i>function j_setfocus (obj: integer) return integer;</i> Directs the input focus to frame obj .
j_setfontname	<i>procedure j_setfontname (obj: integer; name: integer) ;</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Adds a gridlayout manager to frame obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Sets the horizontal gap between components to hgap Pixel.
j_seticon	<i>procedure j_seticon (frame: integer; icon: integer) ;</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: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Removes the current layout manager from frame obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Relocates the frame obj to the specified Position (xpos,ypos).
j_setresizable	<i>procedure j_setresizable (obj: integer; resizable: boolean) ;</i> The frame cannot be resized, if resizable is false .
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Resizes frame obj to specified width and height .
j_settext	<i>procedure j_settext (obj: integer; str: string) ;</i> Sets the content or the label of the frame obj to str .
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) return integer;</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj: integer) ;</i> Shows the frame obj .
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) return integer;</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: integer; columns: integer) return integer;</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) return integer;</i> Creates a new vertical scrollbar and returns its event number.
j_windowlistener	<i>function j_windowlistener (window: integer; kind: integer) return integer;</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: integer) return integer;</i> Creates a new simple window and returns its event number.

Helpmenu

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

Hscrollbar

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

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

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

Graphicbutton

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

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

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

Graphiclabel

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

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

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

Image

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

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

- j_drawrect** *procedure j_drawrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
 Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.
- j_drawroundrect** *procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
 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: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;*
 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: integer; x: integer; y: integer; str: string) ;*
 Draws text on screen at position **(x,y)**.
- j_fillarc** *procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
 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: integer; x: integer; y: integer; r: integer) ;*
 Draws an filled circle with center **(x, y)** and radius **x**.
- j_filloval** *procedure j_filloval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
 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: integer; len: integer; x: integer_array; y: integer_array) ;*
 Draws an filled polygon based on first **len** elements in **x** and **y**.
- j_fillrect** *procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
 Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j_fillroundrect** *procedure j_fillroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
 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: integer) return integer;*
 Returns the height of image **obj**.
- j_getimage** *function j_getimage (obj: integer) return integer;*
 Copy the contents of image **obj** into an image and return its eventnumber.

- j_getimagesource** *function j_getimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: integer_array; g: integer_array; b: integer_array) return integer;*
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: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer) return integer;*
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: integer) return integer;*
Returns the width of image **obj**.
- j_print** *procedure j_print (obj: integer) ;*
prints the image .
- j_setxor** *procedure j_setxor (obj: integer; bool: boolean) ;*
Changes painting mode to XOR mode, if **bool** = 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: integer; x: integer; y: integer) ;*
Moves the origin of drawing operations to (**x**, **y**).

KeyListener

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

Label

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

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

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

Led

j_led	<i>function j_led (obj: integer; style: integer; color: integer) return integer;</i> Creates a new led component with the specified style and the specified color color .
j_add	<i>procedure j_add (obj: integer; cont: integer) ;</i> Adds led obj to container cont
j_componentlistener	<i>function j_componentlistener (obj: integer; kind: integer) return integer;</i> 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	<i>procedure j_disable (obj: integer) ;</i> Disables led obj so that it is unresponsive to user interactions
j_dispose	<i>procedure j_dispose (obj: integer) ;</i> Releases the resources of the led obj .
j_enable	<i>procedure j_enable (obj: integer) ;</i> enables the led obj .
j_focuslistener	<i>function j_focuslistener (obj: integer) return integer;</i> Adds a new focus listener to led obj , and returns its event number.
j_getfontascent	<i>function j_getfontascent (obj: integer) return integer;</i> Returns the ascent (space above the baseline) of the actual font of led obj .
j_getfontheight	<i>function j_getfontheight (obj: integer) return integer;</i> Returns the total pixel height of the actual font of led obj .
j_getheight	<i>function j_getheight (obj: integer) return integer;</i> Returns the height of led obj .
j_getparentid	<i>function j_getparentid (obj: integer) return integer;</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: integer) return integer;</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getstate	<i>function j_getstate (obj: integer) return integer;</i> Returns true , if led is selected, false otherwise.
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: string) return integer;</i> Returns the length of str of the actual font of led obj .
j_getwidth	<i>function j_getwidth (obj: integer) return integer;</i>

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

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

List

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

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

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

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

j_setsize

procedure j_setsize (obj: integer; width: integer; height: integer) ;
Resizes list **obj** to specified **width** and **height**.

j_show

procedure j_show (obj: integer) ;
Shows the list **obj**.

Menu

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

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

j_setshortcut

procedure j_setshortcut (obj: integer; chr: character) ;
Changes the shortcut **chr** of the menu .

j_settext

procedure j_settext (obj: integer; str: string) ;
Sets the content or the label of the menu **obj** to **str**.

Menuitem

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

Meter

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

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

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

Mouselistener

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

Panel

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

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

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

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

j_setgridlayout	<i>procedure j_setgridlayout (obj: integer; row: integer; col: integer) ;</i> Adds a gridlayout manager to panel obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (obj: integer; hgap: integer) ;</i> Sets the horizontal gap between components to hgap Pixel.
j_setinsets	<i>procedure j_setinsets (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;</i> Set the insets to the specified values.
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (obj: integer; color: integer) ;</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (obj: integer; color: integer) ;</i> Sets the foreground color to a predefined color .
j_setnolayout	<i>procedure j_setnolayout (obj: integer) ;</i> Removes the current layout manager from panel obj .
j_setpos	<i>procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;</i> Relocates the panel obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (obj: integer; width: integer; height: integer) ;</i> Resizes panel obj to specified width and height .
j_setvgap	<i>procedure j_setvgap (obj: integer; vgap: integer) ;</i> Sets the vertical gap between components to hgap Pixel.
j_sevensegment	<i>function j_sevensegment (obj: integer; color: integer) return integer;</i> Creates a new sevensegment display with the specified color color .
j_show	<i>procedure j_show (obj: integer) ;</i> Shows the panel obj .
j_textarea	<i>function j_textarea (obj: integer; rows: integer; columns: integer) return integer;</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: integer; columns: integer) return integer;</i> Creates a new textfield component with the specified number of columns and returns its event number.
j_vscrollbar	<i>function j_vscrollbar (obj: integer) return integer;</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: integer; label: string) return integer;</i> Creates a new popupmenu with the specified label and returns its event number.</p>
j_checkmenuitem	<p><i>function j_checkmenuitem (obj: integer; label: string) return integer;</i> creates a new checkmenuitem with the specified label and returns its event number.</p>
j_disable	<p><i>procedure j_disable (obj: integer) ;</i> Disables popupmenu obj so that it is unresponsive to user interactions</p>
j_dispose	<p><i>procedure j_dispose (obj: integer) ;</i> Releases the resources of the popupmenu obj.</p>
j_enable	<p><i>procedure j_enable (obj: integer) ;</i> enables the popupmenu obj.</p>
j_getlength	<p><i>function j_getlength (obj: integer) return integer;</i> Returns the length of popupmenu 's label or text.</p>
j_gettext	<p><i>function j_gettext (obj: integer; str: string) return string;</i> returns the popupmenu 's text or label.</p>
j_menuitem	<p><i>function j_menuitem (obj: integer; label: string) return integer;</i> Creates a new menuitem with the specified label and returns its event number.</p>
j_seperator	<p><i>procedure j_seperator (obj: integer) ;</i> Adds a separator bar to the popupmenu .</p>
j_setfontname	<p><i>procedure j_setfontname (obj: integer; name: integer) ;</i> Changes the font to the given name.</p>
j_setfont	<p><i>procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;</i> Changes the font to the given characteristics name, style and size.</p>
j_setfontsize	<p><i>procedure j_setfontsize (obj: integer; size: integer) ;</i> Changes the font to the given size.</p>
j_setfontstyle	<p><i>procedure j_setfontstyle (obj: integer; style: integer) ;</i> Changes the font to the given style.</p>
j_setshortcut	<p><i>procedure j_setshortcut (obj: integer; chr: character) ;</i> Changes the shortcut chr of the popupmenu .</p>
j_settext	<p><i>procedure j_settext (obj: integer; str: string) ;</i> Sets the content or the label of the popupmenu obj to str.</p>

j_showpopup

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

Printer

- j_printer** *function j_printer (frame: integer) return integer;*
Creates a new object, representing a paper of the printer.
- j_cliprect** *procedure j_cliprect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
Changes current clipping region to the specified rectangle (**x**, **y**, **width**, **height**).
- j_dispose** *procedure j_dispose (obj: integer) ;*
Releases the resources of the printer **obj**.
- j_drawarc** *procedure j_drawarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
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: integer; x: integer; y: integer; r: integer) ;*
Draws an unfilled circle with center (**x**, **y**) and radius **x**.
- j_drawimage** *procedure j_drawimage (obj: integer; image: integer; x: integer; y: integer) ;*
Copies the image, given by its eventnumber **image**, to position (**x**, **y**).
- j_drawimagesource** *procedure j_drawimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: integer_array; g: integer_array; b: integer_array) ;*
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: integer; x1: integer; y1: integer; x2: integer; y2: integer) ;*
Draws a line connecting (**x1,y1**) and (**x2,y2**).
- j_drawoval** *procedure j_drawoval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
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: integer; x: integer; y: integer) ;*
Draws a pixel at (**x,y**).
- j_drawpolygon** *procedure j_drawpolygon (obj: integer; len: integer; x: integer_array; y: integer_array) ;*
Draws an unfilled polygon based on first **len** elements in **x** and **y**.
- j_drawpolyline** *procedure j_drawpolyline (obj: integer; len: integer; x: integer_array; y: integer_array) ;*
Draws a series of line segments based on first **len** elements in **x** and **y**.

- j_drawrect** *procedure j_drawrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
 Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.
- j_drawroundrect** *procedure j_drawroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
 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: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;*
 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: integer; x: integer; y: integer; str: string) ;*
 Draws text on screen at position **(x,y)**.
- j_fillarc** *procedure j_fillarc (obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer) ;*
 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: integer; x: integer; y: integer; r: integer) ;*
 Draws an filled circle with center **(x, y)** and radius **x**.
- j_filloval** *procedure j_filloval (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;*
 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: integer; len: integer; x: integer_array; y: integer_array) ;*
 Draws an filled polygon based on first **len** elements in **x** and **y**.
- j_fillrect** *procedure j_fillrect (obj: integer; x: integer; y: integer; width: integer; height: integer) ;*
 Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j_fillroundrect** *procedure j_fillroundrect (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer) ;*
 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: integer) ;*
 prints the printer .
- j_setxor** *procedure j_setxor (obj: integer; bool: boolean) ;*
 Changes painting mode to XOR mode, if **bool = 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: integer; x: integer; y: integer) ;
Moves the origin of drawing operations to (**x**, **y**).

Progressbar

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

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

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

Radiobutton

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

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

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

Sevensegment

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

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

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

Scrollpane

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

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

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

Textarea

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

j_getlength	<i>function j_getlength (obj: integer) return integer;</i> Returns the length of textarea 's label or text.
j_getparentid	<i>function j_getparentid (obj: integer) return integer;</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: integer) return integer;</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: integer) ;</i> Gets the number of rows in obj .
j_getselend	<i>function j_getselend (obj: integer) return integer;</i> Returns the ending position of any selected text.
j_getselstart	<i>function j_getselstart (obj: integer) return integer;</i> Returns the initial position of any selected text.
j_getseltext	<i>function j_getseltext (obj: integer; text: string) return string;</i> Returns the currently selected text of textarea obj .
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: string) return integer;</i> Returns the length of str of the actual font of textarea obj .
j_gettext	<i>function j_gettext (obj: integer; str: string) return string;</i> returns the textarea 's text or label.
j_getwidth	<i>function j_getwidth (obj: integer) return integer;</i> Returns the width of textarea obj .
j_getxpos	<i>function j_getxpos (obj: integer) return integer;</i> Returns the current horizontal position of textarea obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj: integer) return integer;</i> Returns the current vertical position of textarea obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Hides the textarea obj .
j_inserttext	<i>procedure j_inserttext (obj: integer; text: string; pos: integer) ;</i> Places additional text within the textarea at the given position pos .
j_isparent	<i>function j_isparent (obj: integer; cont: integer) return integer;</i> Returns true if cont is parent of obj , false otherwise.
j_isvisible	<i>function j_isvisible (obj: integer) return integer;</i> Returns true if obj is visible, false otherwise.
j_keylistener	<i>function j_keylistener (obj: integer) return integer;</i>

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

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

Textfield

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

j_getparent	<i>function j_getparent (obj: integer) return integer;</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: integer) return integer;</i> Returns the ending position of any selected text.
j_getselstart	<i>function j_getselstart (obj: integer) return integer;</i> Returns the initial position of any selected text.
j_getseltext	<i>function j_getseltext (obj: integer; text: string) return string;</i> Returns the currently selected text of textfield obj .
j_getstringwidth	<i>function j_getstringwidth (obj: integer; str: string) return integer;</i> Returns the length of str of the actual font of textfield obj .
j_gettext	<i>function j_gettext (obj: integer; str: string) return string;</i> returns the textfield 's text or label.
j_getwidth	<i>function j_getwidth (obj: integer) return integer;</i> Returns the width of textfield obj .
j_getxpos	<i>function j_getxpos (obj: integer) return integer;</i> Returns the current horizontal position of textfield obj in its parent's coordinate space.
j_getypos	<i>function j_getypos (obj: integer) return integer;</i> Returns the current vertical position of textfield obj in its parent's coordinate space.
j_hide	<i>procedure j_hide (obj: integer) ;</i> Hides the textfield obj .
j_isparent	<i>function j_isparent (obj: integer; cont: integer) return integer;</i> Returns true if cont is parent of obj , false otherwise.
j_isvisible	<i>function j_isvisible (obj: integer) return integer;</i> Returns true if obj is visible, false otherwise.
j_keylistener	<i>function j_keylistener (obj: integer) return integer;</i> Adds a new key listener to textfield obj , and returns its event number.
j_mouselistener	<i>function j_mouselistener (obj: integer; kind: integer) return integer;</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: integer; label: string) return integer;</i> Creates a new popupmenu with the specified label and returns its event number.
j_print	<i>procedure j_print (obj: integer) ;</i> prints the textfield .

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

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

Vscrollbar

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

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

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

Window

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

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

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

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

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

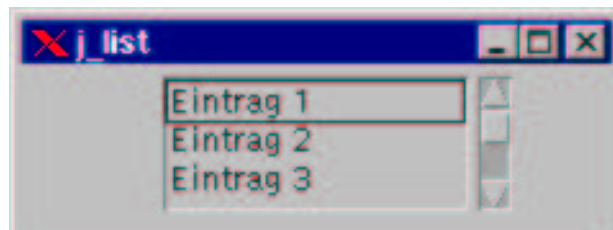
Arguments obj integer
 str string

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	procedure j_add (obj: integer; cont: integer) ;
Arguments	obj integer cont integer
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, Sevenssegment

alertbox

Synopsis procedure **j_alertbox** (obj: integer; title: string; text: string;
 button: string) ;

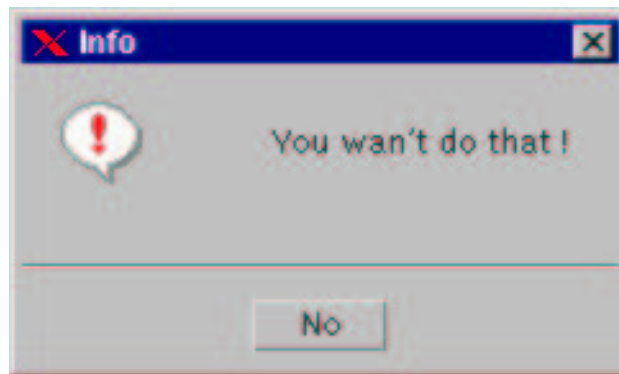
Arguments obj integer
 title string
 text string
 button string

Description Shows a alertbox with the specified **title**, **text** and **button**. Alert-
boxes are modal dialogs, the application is blocked until the but-
ton 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 procedure **j_appendtext** (obj: integer; text: string) ;

Arguments obj integer
 text string

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

Targets Textarea

beep

Synopsis procedure **j_beep** ;

Description Emits an audio beep.

borderpanel

Synopsis function **j_borderpanel** (obj: integer; type: integer) return integer;

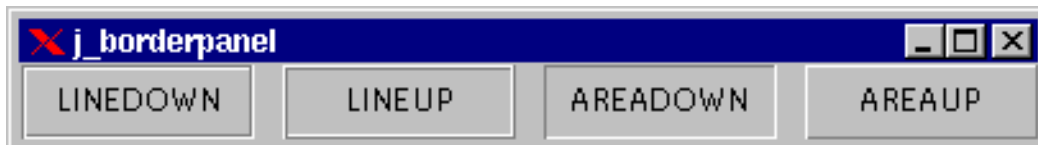
Arguments obj integer
 type integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

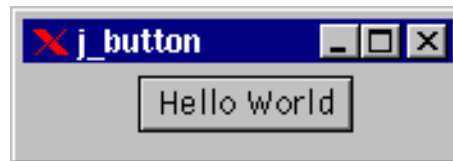
Example

```
:  
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: integer; label: string) return integer;
Arguments	obj integer label string
Description	Creates a new button component with the specified label and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: frame := j_frame('j_button'); button := j_button(frame,'Hello World'); :</pre>



canvas

Synopsis function **j_canvas** (obj: integer; width: integer; height: integer
) return integer;

Arguments obj integer
 width integer
 height integer

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: integer; label: string) return integer;`

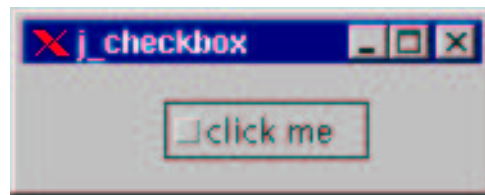
Arguments `obj integer`
 `label string`

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: integer; label: string) re-
return integer;

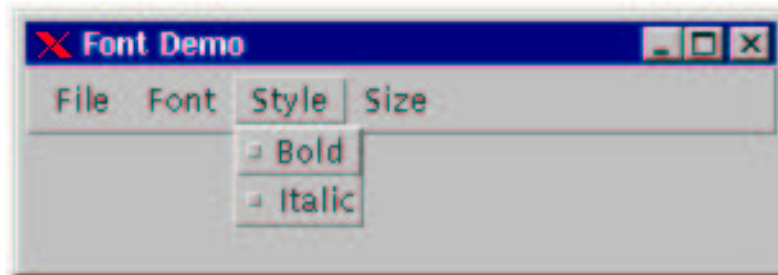
Arguments obj integer
 label string

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');  
:
```



checkbox2

Synopsis procedure **j_checkbox2** (obj: integer; title: string; text: string;
 button1: string; button2: string) ;

Arguments

obj	integer
title	string
text	string
button1	string
button2	string

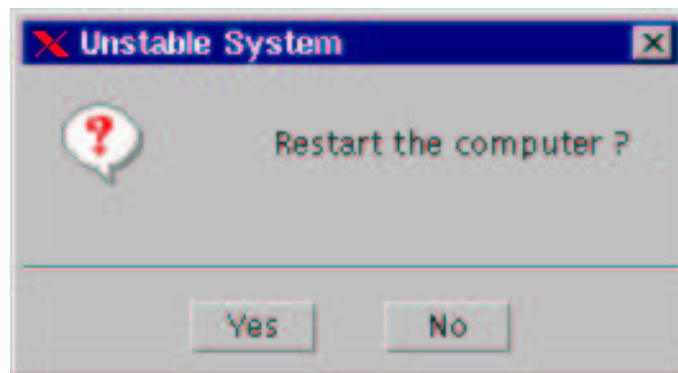
Description

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_checkbox2(frame,'Unstable System','Restart the computer ?',  
                         'Yes','No');  
:
```



checkbox3

Synopsis `procedure j_checkbox3 (obj: integer; title: string; text: string;
button1: string; button2: string; button3: string) ;`

Arguments

<code>obj</code>	<code>integer</code>
<code>title</code>	<code>string</code>
<code>text</code>	<code>string</code>
<code>button1</code>	<code>string</code>
<code>button2</code>	<code>string</code>
<code>button3</code>	<code>string</code>

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.

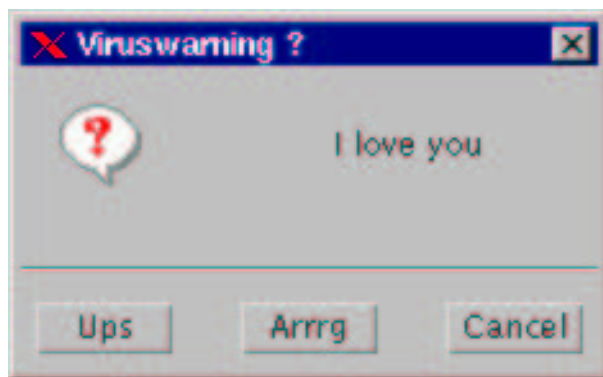
Targets Frame

Example

```

:
retval := j_checkbox2(frame,'Viruswarning ??','I love you',
                        'Ups','Arrrg','Cancel');
:

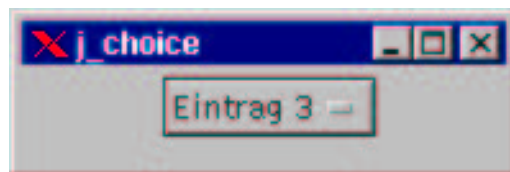
```



choice

Synopsis	function j_choice (obj: integer) return integer;
Arguments	obj integer
Description	Creates a new choice component and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	

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



cliprect

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

componentlistener

Synopsis	function j_componentlistener (obj: integer; kind: integer) return integer;
Arguments	obj integer kind integer
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: string) return boolean;

Arguments hostname string

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

Example

```
:
if( not j_connect('atan.japi.de') ) then

    or

if( not j_connect('127.0.0.1')) then
:
```


delete

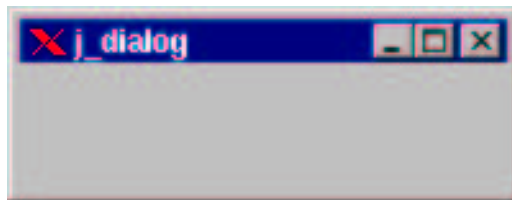
Synopsis	procedure j_delete (obj: integer; start: integer; end: integer) ;
Arguments	obj integer start integer end integer
Description	Deletes text from starting position start to ending position end .
Targets	Textarea

deselect

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

dialog

Synopsis	function j_dialog (obj: integer; label: string) return integer;
Arguments	obj integer label string
Description	Creates a new dialog window with the specified label and returns its event number.
Targets	Frame
Example	<pre>: dialog := j_dialog(frame,'j_dialog'); j_setsize(dialog,200,80); j_show(dialog); :</pre>



disable

Synopsis	procedure j_disable (obj: integer) ;
Arguments	obj integer
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, Popupmenu

dispose

Synopsis	procedure j_dispose (obj: integer) ;
Arguments	obj integer
Description	Releases the resources of 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, Sevenssegment, Canvas, Image, Printer, Keylistener, Focuslistener, Mouselistener

drawarc

Synopsis procedure **j_drawarc** (obj: integer; x: integer; y: integer; rx:
integer; ry: integer; arc1: integer; arc2: integer) ;

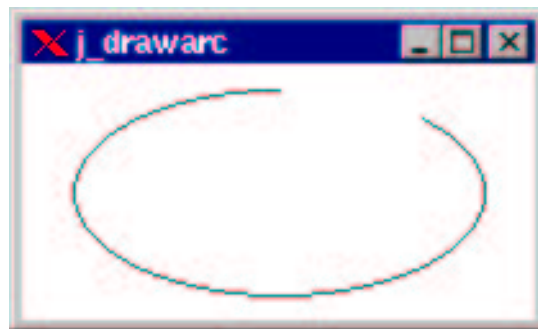
Arguments obj integer
 x integer
 y integer
 rx integer
 ry integer
 arc1 integer
 arc2 integer

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 procedure **j_drawcircle** (obj: integer; x: integer; y: integer; r:
integer) ;

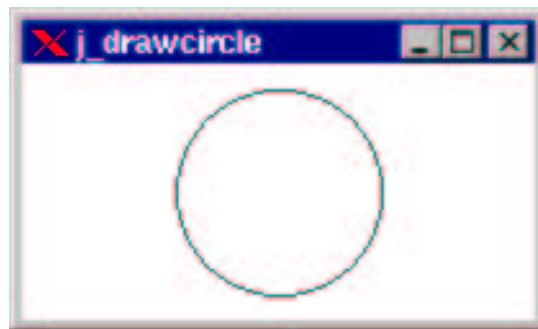
Arguments obj integer
 x integer
 y integer
 r integer

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	<pre>procedure j_drawimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: integer_array; g: integer_array; b: integer_array) ;</pre>																
Arguments	<table><tr><td>obj</td><td>integer</td></tr><tr><td>x</td><td>integer</td></tr><tr><td>y</td><td>integer</td></tr><tr><td>w</td><td>integer</td></tr><tr><td>h</td><td>integer</td></tr><tr><td>r</td><td>integer_array</td></tr><tr><td>g</td><td>integer_array</td></tr><tr><td>b</td><td>integer_array</td></tr></table>	obj	integer	x	integer	y	integer	w	integer	h	integer	r	integer_array	g	integer_array	b	integer_array
obj	integer																
x	integer																
y	integer																
w	integer																
h	integer																
r	integer_array																
g	integer_array																
b	integer_array																
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	procedure j_drawimage (obj: integer; image: integer; x: integer; y: integer) ;
Arguments	obj integer image integer x integer y integer
Description	Copies the image, given by its eventnumber image , to position (x , y).
Targets	Canvas, Image, Printer

drawline

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

Arguments obj integer
 x1 integer
 y1 integer
 x2 integer
 y2 integer

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 procedure **j_drawoval** (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;

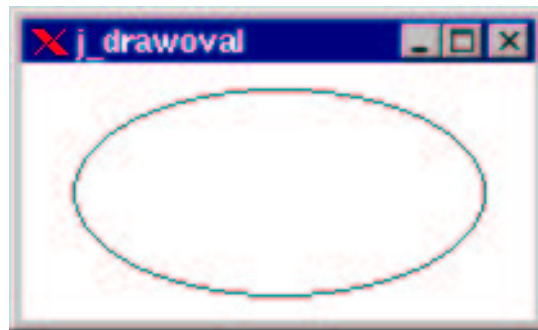
Arguments obj integer
 x integer
 y integer
 rx integer
 ry integer

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 `procedure j_drawpixel (obj: integer; x: integer; y: integer) ;`

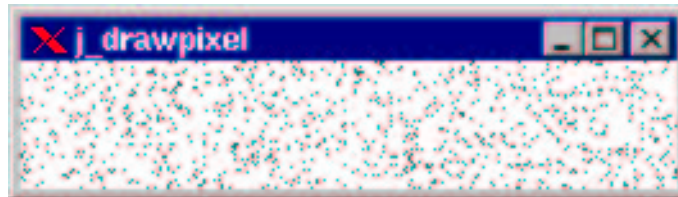
Arguments `obj integer`
 `x integer`
 `y integer`

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

Targets `Canvas, Image, Printer`

Example

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



drawpolygon

Synopsis procedure **j_drawpolygon** (obj: integer; len: integer; x: integer_array; y: integer_array) ;

Arguments

obj	integer
len	integer
x	integer_array
y	integer_array

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

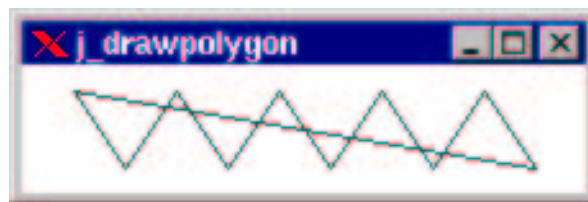
Targets Canvas, Image, Printer

Example

```

:
x : array [0..9] of int =
    (20,40,60,80,100,120,140,160,180,200);
y : array [0..9] of int =
    (10,40,10,40,10,40,10,40,10,40);
canvas := j_canvas(frame,256,50);
j_drawpolygon(canvas,10,x[0],y[0]);
:

```



drawpolyline

Synopsis procedure **j_drawpolyline** (obj: integer; len: integer; x: integer_array; y: integer_array) ;

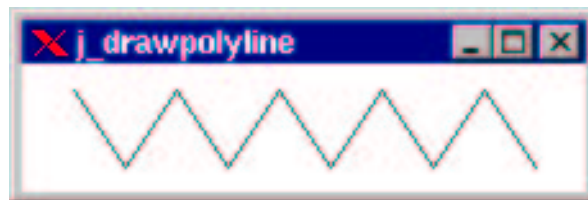
Arguments obj integer
 len integer
 x integer_array
 y integer_array

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

Targets Canvas, Image, Printer

Example

```
:  
x : array [0..9] of int =  
    (20,40,60,80,100,120,140,160,180,200);  
y : array [0..9] of int =  
    (10,40,10,40,10,40,10,40,10,40);  
canvas := j_canvas(frame,256,50);  
j_drawpolyline(canvas,10,x[0],y[0]);  
:
```



drawrect

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

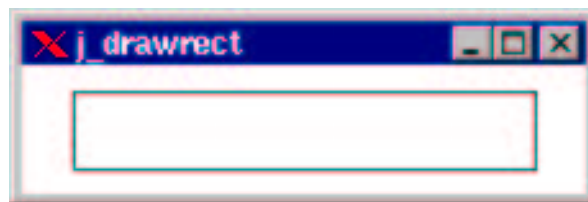
Arguments obj integer
 x integer
 y integer
 width integer
 height integer

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 procedure **j_drawroundrect** (obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer)
;

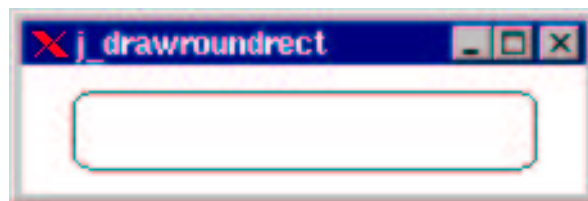
Arguments obj integer
 x integer
 y integer
 width integer
 height integer
 arcx integer
 arcy integer

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

Targets Canvas, Image, Printer

Example

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



drawscaledimage

Synopsis	procedure j_drawscaledimage (obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer) ;
Arguments	obj integer image integer sx integer sy integer sw integer sh integer tx integer ty integer tw integer th integer
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 procedure **j_drawstring** (obj: integer; x: integer; y: integer;
 str: string) ;

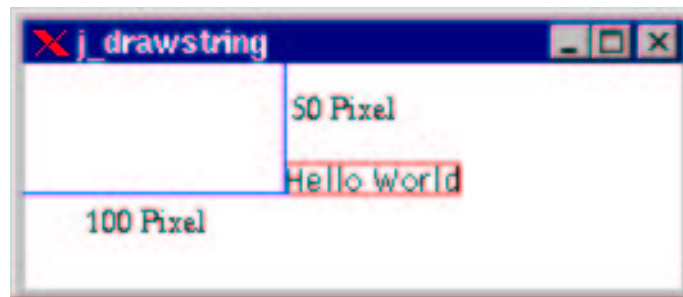
Arguments obj integer
 x integer
 y integer
 str string

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

Targets Canvas, Image, Printer

Example

```
:  
j_drawstring(canvas,100,100,'Hello World');  
:
```

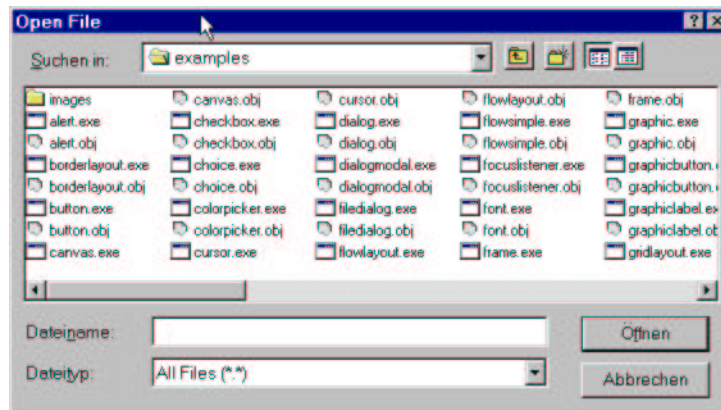


enable

Synopsis	procedure j_enable (obj: integer) ;
Arguments	obj integer
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, Sevensgment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

filedialog

Synopsis	function j_filedialog (frame: integer; title: string; directory: string; filename: string) return string;								
Arguments	<table border="0"> <tr><td>frame</td><td>integer</td></tr> <tr><td>title</td><td>string</td></tr> <tr><td>directory</td><td>string</td></tr> <tr><td>filename</td><td>string</td></tr> </table>	frame	integer	title	string	directory	string	filename	string
frame	integer								
title	string								
directory	string								
filename	string								
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	<pre> : filename := j_filedialog(frame,'Save/S File','..',filename); : </pre>								



fileselect

Synopsis function **j_fileselect** (frame: integer; title: string; filter: string;
filename: string) return string;

Arguments

frame	integer
title	string
filter	string
filename	string

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 `j_connect()`;

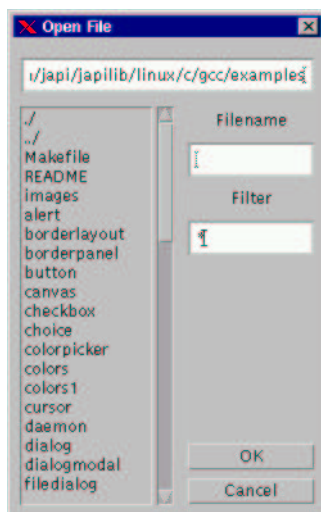
Targets Frame

Example

```

:
filename := j_fileselect(frame,'Open File','*',filename);
:

```



fillarc

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

Arguments

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

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 procedure **j_fillcircle** (obj: integer; x: integer; y: integer; r:
integer) ;

Arguments obj integer
 x integer
 y integer
 r integer

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 procedure **j_filloval** (obj: integer; x: integer; y: integer; rx: integer; ry: integer) ;

Arguments obj integer
 x integer
 y integer
 rx integer
 ry integer

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 procedure **j_fillpolygon** (obj: integer; len: integer; x: integer_array; y: integer_array);

Arguments

obj	integer
len	integer
x	integer_array
y	integer_array

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

Targets Canvas, Image, Printer

Example

```

:
x : array [0..9] of int =
    (20,40,60,80,100,120,140,160,180,200);
y : array [0..9] of int =
    (10,40,10,40,10,40,10,40,10,40);
canvas := j_canvas(frame,256,50);
j_fillpolygon(canvas,10,x[0],y[0]);
:

```



fillrect

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

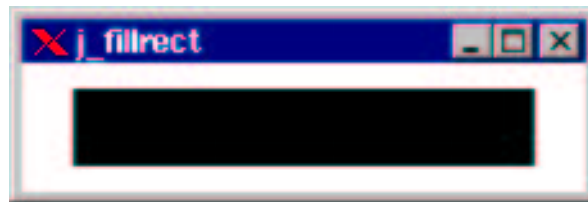
Arguments obj integer
 x integer
 y integer
 width integer
 height integer

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

Arguments

obj	integer
x	integer
y	integer
width	integer
height	integer
arcx	integer
arcy	integer

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

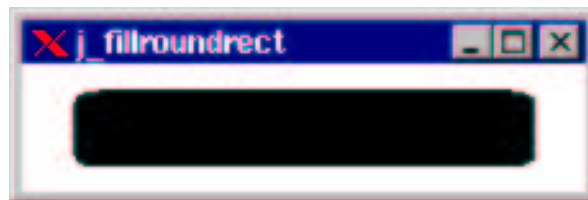
Targets Canvas, Image, Printer

Example

```

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

```



focuslistener

Synopsis	function j_focuslistener (obj: integer) return integer;
Arguments	obj integer
Description	Adds a new focus listener to component obj , and returns its event number.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

frame

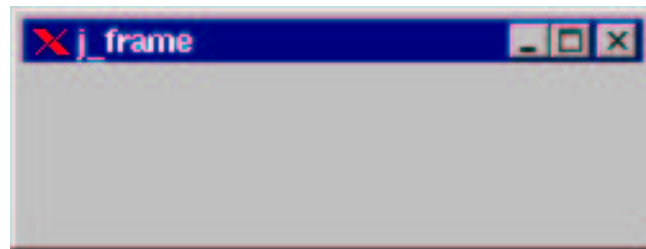
Synopsis function **j_frame** (label: string) return integer;

Arguments label string

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** return integer;

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

getcolumns

Synopsis `procedure j_getcolumns (obj: integer) ;`

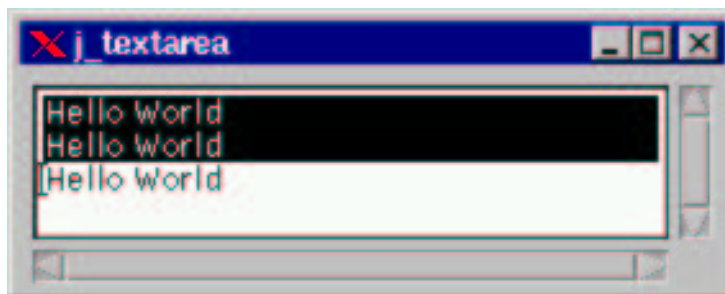
Arguments `obj` `integer`

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: integer) return integer;

Arguments obj integer

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

Targets Textarea, Textfield

getdanger

Synopsis `procedure j_getdanger (obj: integer) ;`

Arguments `obj integer`

Description Returns the danger value of component **obj**.

Targets Meter

getfontascent

Synopsis	function j_getfontascent (obj: integer) return integer;
Arguments	obj integer
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, Sevenssegment

getfontheight

Synopsis	function j_getfontheight (obj: integer) return integer;
Arguments	obj integer
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: integer) return integer;

Arguments obj integer

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');  
writeln(j_getheight(label));  
:  
  
> 22
```

getimagesource

Synopsis	function j_getimagesource (obj: integer; x: integer; y: integer; w: integer; h: integer; r: integer_array; g: integer_array; b: integer_array) return integer;																
Arguments	<table><tr><td>obj</td><td>integer</td></tr><tr><td>x</td><td>integer</td></tr><tr><td>y</td><td>integer</td></tr><tr><td>w</td><td>integer</td></tr><tr><td>h</td><td>integer</td></tr><tr><td>r</td><td>integer_array</td></tr><tr><td>g</td><td>integer_array</td></tr><tr><td>b</td><td>integer_array</td></tr></table>	obj	integer	x	integer	y	integer	w	integer	h	integer	r	integer_array	g	integer_array	b	integer_array
obj	integer																
x	integer																
y	integer																
w	integer																
h	integer																
r	integer_array																
g	integer_array																
b	integer_array																
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: integer) return integer;
Arguments	obj integer
Description	Copy the contents of component obj into an image and return its eventnumber.
Targets	Canvas, Image

getinheight

Synopsis function **j_getinheight** (cont: integer) return integer;

Arguments cont integer

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);  
writeln(j_getinheight(frame));  
:  
> 370
```

getinsets

Synopsis function **j_getinsets** (obj: integer; side: integer) return integer;

Arguments obj integer
 side integer

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');
writeln(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: integer) return integer;

Arguments cont integer

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);
writeln(j_getinwidth(frame));
:
> 289
```

getitemcount

Synopsis function **j_getitemcount** (obj: integer) return integer;

Arguments obj integer

Description Returns the number of items of component **obj**.

Targets List, Choice

getitem

Synopsis	function j_getitem (obj: integer; item: integer; str: string) return string;
Arguments	obj integer item integer str string
Description	returns the label of the given item .
Targets	List, Choice

getkeychar

Synopsis function **j_getkeychar** (obj: integer) return integer;

Arguments obj integer

Description Returns the ascii value of the last pressed key.

Targets Keylistener

getkeycode

Synopsis function **j_getkeycode** (obj: integer) return integer;

Arguments obj integer

Description Returns the integer key code of the last pressed key.

Targets Keylistener

getlayoutid

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

getmousex

Synopsis	function j_getmousex (mouset listener: integer) return integer;
Arguments	mouset listenerinteger
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: integer) return integer;
Arguments	mouset listenerinteger
Description	Returns the current vertical position of the mouse in its parent's coordinate space.
Targets	Mouset listener

getparentid

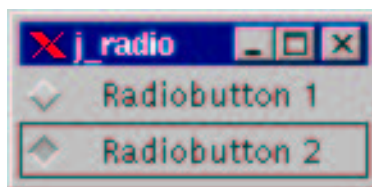
Synopsis	function j_getparentid (obj: integer) return integer;
Arguments	obj integer
Description	Returns the parent event number of component obj . If obj is a frame -1 will be returned.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel,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

Example

```

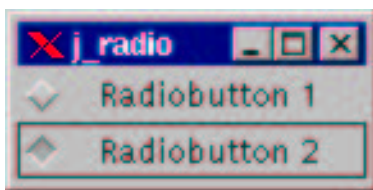
:
radio1        := j_radiobutton(j_radiogroup(frame), 'Radiobutton 1');
radio2        := j_radiobutton(j_getparentid(radio1), 'Radiobutton 2');
:

```



getparent

Synopsis	function j_getparent (obj: integer) return integer;
Arguments	obj integer
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	<pre>: radio1 := j_radiobutton(j_radiogroup(frame), 'Radiobutton 1'); radio2 := j_radiobutton(j_getparent(radio1), 'Radiobutton 2'); :</pre>



getrows

Synopsis procedure **j_getrows** (obj: integer) ;

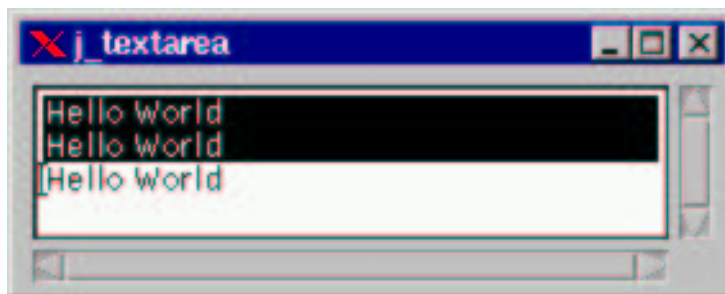
Arguments obj integer

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: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer) return integer;
Arguments	obj integer x integer y integer sw integer sh integer tw integer th integer
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** return integer;

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

Example

```
:  
writeln(j_getscreenwidth(), j_getscreenheight());  
:  
  
> 1280 1024
```

getscreenwidth

Synopsis function **j_getscreenwidth** return integer;

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

Example

```
:  
writeln(j_getscreenwidth(), j_getscreenheight());  
:  
  
> 1280 1024
```


getselect

Synopsis function **j_getselect** (obj: integer) return integer;

Arguments obj integer

Description Returns the position of currently selected item.

Targets List, Choice

getselend

Synopsis function **j_getselend** (obj: integer) return integer;

Arguments obj integer

Description Returns the ending position of any selected text.

Targets Textarea, Textfield

getselstart

Synopsis function **j_getselstart** (obj: integer) return integer;

Arguments obj integer

Description Returns the initial position of any selected text.

Targets Textarea, Textfield

getseltext

Synopsis `function j_getseltext (obj: integer; text: string) return string;`

Arguments `obj` `integer`
 `text` `string`

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

Targets Textarea, Textfield

getstate

Synopsis	function j_getstate (obj: integer) return integer;
Arguments	obj integer
Description	Returns true , if component is selected, false otherwise.
Targets	Checkbox, Radiobutton, Checkmenuitem, Led

getstringwidth

Synopsis	function j_getstringwidth (obj: integer; str: string) return integer;
Arguments	obj integer str string
Description	Returns the length of str of the actual font of 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

gettext

Synopsis	function j_gettext (obj: integer; str: string) return string;
Arguments	obj integer str string
Description	returns the component 's text or label.
Targets	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield
Example	<pre>str : PChar; : getmem(str,256); label := j_label(frame,'Hello World'); writeln(j_gettext(label,str)); : > Hello World</pre>

getvalue

Synopsis function **j_getvalue** (obj: integer) return integer;

Arguments obj integer

Description Returns the current setting of the scrollbar.

Targets Scrollbar

getviewportheight

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

getviewportwidth

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

getwidth

Synopsis function **j_getwidth** (obj: integer) return integer;

Arguments obj integer

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

```
:
label := j_getlabel(frame, 'Hello World');
writeln(j_getwidth(label));
:
> 84
```

getxpos

Synopsis	function j_getxpos (obj: integer) return integer;
Arguments	obj integer
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, Sevenssegment

getypos

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

graphicbutton

Synopsis `function j_graphicbutton (obj: integer; filename: string) re-`
`turn integer;`

Arguments `obj` `integer`
 `filename` `string`

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: integer; str: string) return integer;

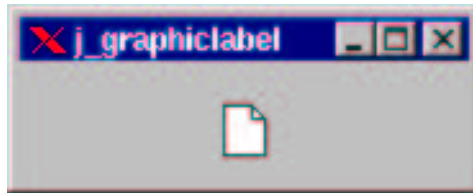
Arguments obj integer
 str string

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: integer) return integer;

Arguments obj integer

Description Returns true if the component has the focus, false otherwise.

Targets Focuslistener

helpmenu

Synopsis `function j_helpmenu (obj: integer; label: string) return integer;`

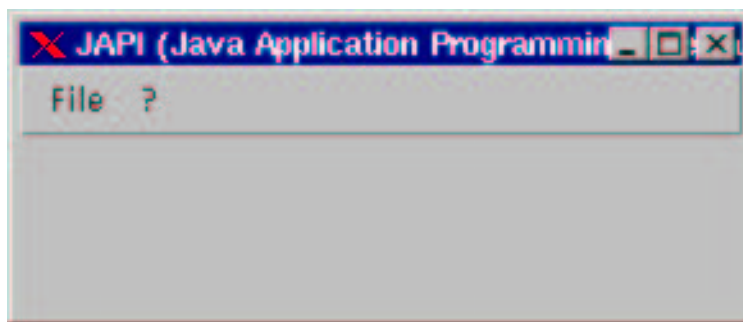
Arguments `obj integer`
`label string`

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 procedure **j_hide** (obj: integer) ;

Arguments obj integer

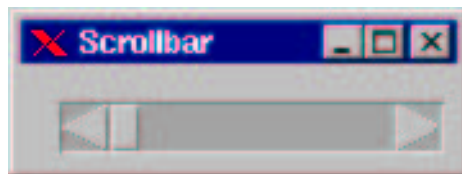
Description Hides the component **obj**.

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: integer) return integer;
Arguments	obj integer
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: integer; height: integer) return integer;

Arguments width integer
 height integer

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 j_insert (obj: integer; pos: integer; label: string) return integer;
Arguments	obj integer pos integer label string
Description	inserts a new item to component obj at position pos with the specified label .
Targets	List, Choice

inserttext

Synopsis	procedure j_inserttext (obj: integer; text: string; pos: integer) ;
Arguments	obj integer text string pos integer
Description	Places additional text within the component at the given position pos .
Targets	Textarea

isparent

Synopsis	function j_isparent (obj: integer; cont: integer) return integer;
Arguments	obj integer cont integer
Description	Returns true if cont is parent of obj , false otherwise.
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

isresizable

Synopsis function **j_isresizable** (obj: integer) return boolean;

Arguments obj integer

Description returns true if component is resizable, false otherwise

Targets Dialog, Frame

isselect

Synopsis	function j_isselect (obj: integer; item: integer) return integer;
Arguments	obj integer item integer
Description	Returns true if the particular item is currently selected, false otherwise.
Targets	List

isvisible

Synopsis function **j_isvisible** (obj: integer) return integer;

Arguments obj integer

Description Returns true if **obj** is visible, 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

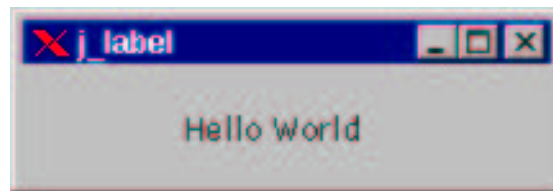
keylistener

Synopsis	function j_keylistener (obj: integer) return integer;
Arguments	obj integer
Description	Adds a new key listener to component obj , and returns its event number.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

label

Synopsis	function j_label (obj: integer; label: string) return integer;
Arguments	obj integer label string
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: integer; style: integer; color: integer) return integer;

Arguments

obj	integer
style	integer
color	integer

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: integer; orient: integer; style: integer;
length: integer) return integer;

Arguments obj integer
 orient integer
 style integer
 length integer

Description Creates a new line component with the specified **length** and re-
turns its event number. A line may be used to separate groups of
components. On Error `-1` will returned. The parameter **orient**
specifies the orientation of the line:

- J_HORIZONTAL : horizontal line
- J_VERTICAL : vertical line

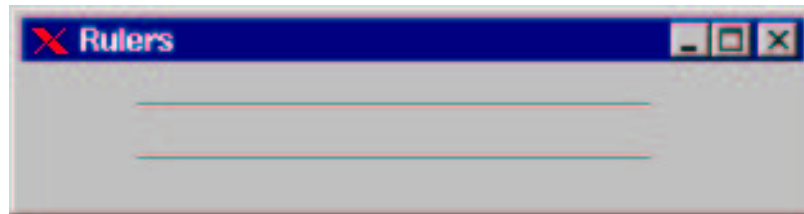
The Parameter **style** specifies the linestyle:

- J_LINEDOWN : etched-in linestyle.
- J_LINEUP : etchet-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);
:
```

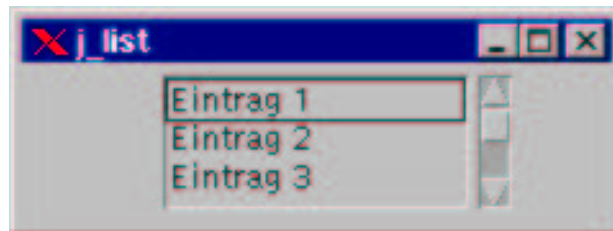


list

Synopsis	function j_list (obj: integer; rows: integer) return integer;
Arguments	obj integer rows integer
Description	Creates a new list component with the specified number of rows 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: string) return integer;
Arguments	filename string
Description	Loads the Image from file filename and returns its eventnumber. The file could be of the following format: <ul style="list-style-type: none">• GIF• JPEG• BMP• PPM
Example	<pre>: image := j_loadimage("mandel.jpg"); :</pre>

menubar

Synopsis function **j_menubar** (obj: integer) return integer;

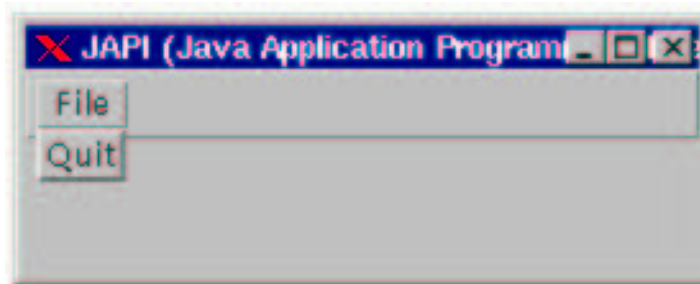
Arguments obj integer

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: integer; label: string) return integer;`

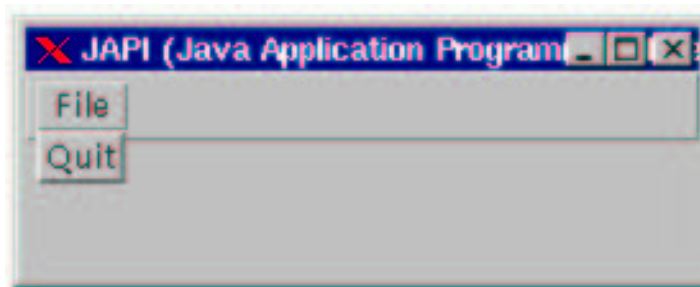
Arguments `obj integer`
`label string`

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

Targets Menu, Popupmenu, Helpmenu

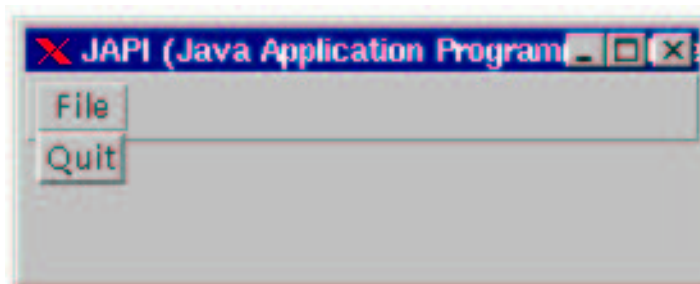
Example

```
:  
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: integer; str: string) return integer;
Arguments	obj integer str string
Description	Creates a new menu component with the specified label and returns its event number.
Targets	Menubar, Menu
Example	<pre>: frame := j_frame('Menu Komponenten'); menubar := j_menubar(frame); file := j_menu(menubar,'File'); quit := j_menuitem(file,'Quit'); :</pre>



messagebox

Synopsis procedure **j_messagebox** (obj: integer; title: string; text:
 string) ;

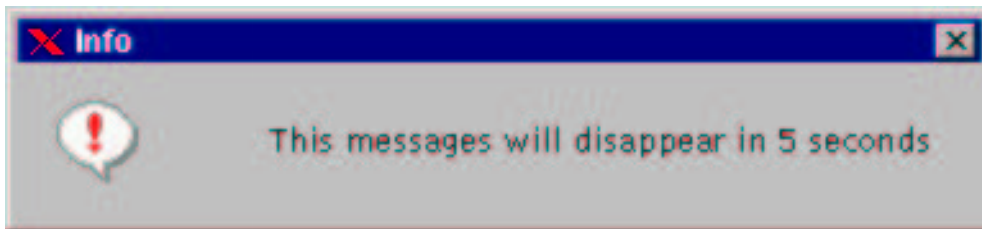
Arguments obj integer
 title string
 text string

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

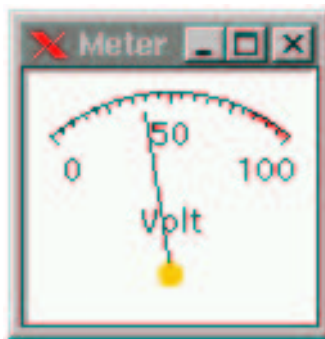
Example

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



meter

Synopsis	function j_meter (obj: integer; title: string) return integer;
Arguments	obj integer title string
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 <code>j_setmin()</code> and <code>j_setmax()</code> . A danger value is set to 80 and can be justified with <code>j_setdanger()</code> .
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre> : meter := j_meter(frame,"Volt"); j_setvalue(meter,40); : </pre>



mouselistener

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

multiplemode

Synopsis	function j_multiplemode (obj: integer; bool: boolean) return integer;
Arguments	obj integer bool boolean
Description	if bool is true , selection mode is turned to multiplemode.
Targets	List

nextaction

Synopsis function **j_nextaction** return integer;

Description Waits for the next event.

pack

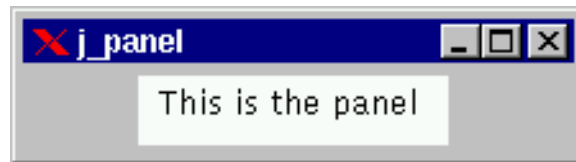
Synopsis	procedure j_pack (obj: integer) ;
Arguments	obj integer
Description	Resizes component to the minimal size of contained components.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: j_setflowlayout(jframe,J_HORIZONTAL); canvas := j_canvas(frame,200,50); j_setnamedcolorbg(canvas,J_RED);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



panel

Synopsis	function j_panel (obj: integer) return integer;
Arguments	obj integer
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 **j_popupmenu** (obj: integer; label: string) return integer;

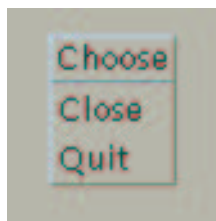
Arguments obj integer
 label string

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

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

Example

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



printer

Synopsis function **j_printer** (frame: integer) return integer;

Arguments frame integer

Description Creates a new object, representing a paper of the printer and returns its event number. On error -1 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

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

print

Synopsis	procedure j_print (obj: integer) ;
Arguments	obj integer
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, Sevenssegment, Canvas, Image, Printer
Example	<pre> : frame := j_frame('j_textfield'); text := j_textfield(frame,30) : j_print(frame); : </pre>



progressbar

Synopsis function **j_progressbar** (obj: integer; orient: integer) return integer;

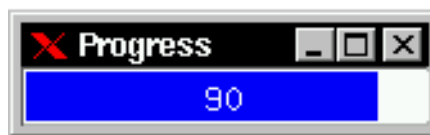
Arguments obj integer
 orient integer

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 `procedure j_quit ;`

Description Cancels the connection to the JAPI Kernel.

radiobutton

Synopsis function **j_radiobutton** (obj: integer; label: string) return
integer;

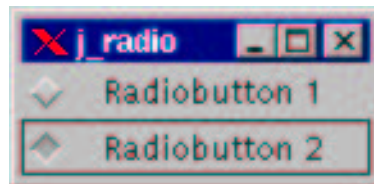
Arguments obj integer
 label string

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: integer) return integer;`

Arguments `obj integer`

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** return integer;

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

releaseall

Synopsis	procedure j_releaseall (obj: integer) ;
Arguments	obj integer
Description	Releases all components from component obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame

release

Synopsis	procedure j_release (obj: integer) ;
Arguments	obj integer
Description	Releases component obj from its parent component (container).
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

removeall

Synopsis function **j_removeall** (obj: integer) return integer;

Arguments obj integer

Description Removes all items from the component .

Targets List, Choice

removeitem

Synopsis	function j_removeitem (obj: integer; item: string) return integer;
Arguments	obj integer item string
Description	remove the first occurrence of item from the component .
Targets	List, Choice

remove

Synopsis function **j_remove** (obj: integer; item: integer) return integer;

Arguments obj integer
 item integer

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

Targets List, Choice

replacetext

Synopsis	procedure j_replacetext (obj: integer; text: string; start: integer; end: integer) ;
Arguments	obj integer text string start integer end integer
Description	Replaces the text from starting position start to ending position end with the given text .
Targets	Textarea

saveimage

Synopsis function **j_saveimage** (obj: integer; filename: string; filetype:
 integer) return boolean;

Arguments obj integer
 filename string
 filetyp integer

Description Saves the components image to file **filename**. The specified file
 format can be:

- J_BMP Win32 Bitmap Format
- J_PPM Portable pixmap

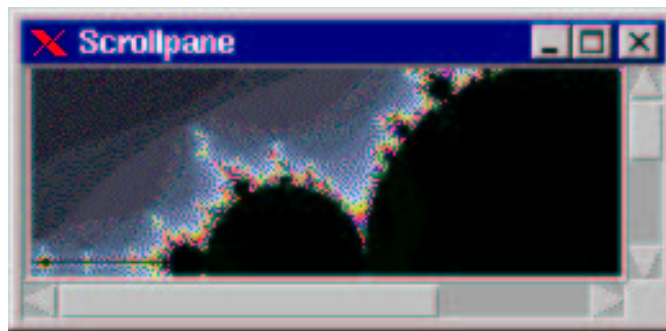
Example

```
:  
if(j_saveimage(canvas,"mandel.bmp",J_BMP) = false)  
  writeln('Error saving Bitmap file');  
:
```

scrollpane

Synopsis	function j_scrollpane (obj: integer) return integer;
Arguments	obj integer
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 procedure **j_selectall** (obj: integer) ;

Arguments obj integer

Description Selects all the text in the component .

Targets Textarea, Textfield

select

Synopsis function **j_select** (obj: integer; item: integer) return integer;

Arguments obj integer
 item integer

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

Targets List, Choice

selecttext

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

seperator

Synopsis `procedure j_seperator (obj: integer) ;`

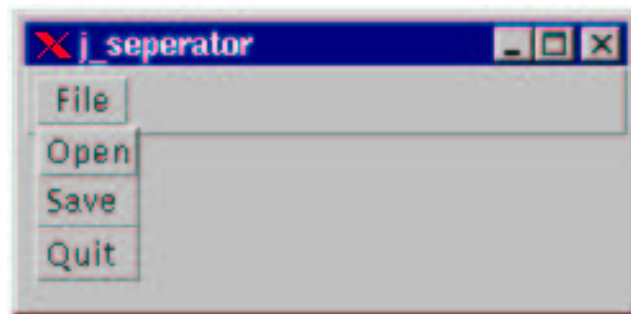
Arguments `obj integer`

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	procedure j_setalign (obj: integer; align: integer) ;
Arguments	obj integer align integer
Description	Sets the alignment in component obj to align . Needs a flowlayout Manager.
Targets	Panel, Borderpanel, Window, Dialog, Frame

setblockinc

Synopsis	function j_setblockinc (obj: integer; val: integer) return integer;
Arguments	obj integer val integer
Description	Changes the block increment amount for the component to val .
Targets	Scrollbar

setborderlayout

Synopsis procedure **j_setborderlayout** (obj: integer) ;

Arguments obj integer

Description Adds a borderlayout manager to component **obj**.

Targets Panel, Borderpanel, Window, Dialog, Frame

setborderpos

Synopsis	procedure j_setborderpos (obj: integer; pos: integer) ;
Arguments	obj integer pos integer
Description	Moves component obj 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 procedure **j_setcolorbg** (obj: integer; r: integer; g,: integer; b:
integer) ;

Arguments obj integer
 r integer
 g, integer
 b integer

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

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice,
Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window,
Dialog, Frame, Scrollpane, Textarea, Textfield, 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 procedure **j_setcolor** (obj: integer; r: integer; g,: integer; b:
integer) ;


Arguments obj integer
 r integer
 g, integer
 b integer

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 `procedure j_setcolumns (obj: integer; columns: integer) ;`

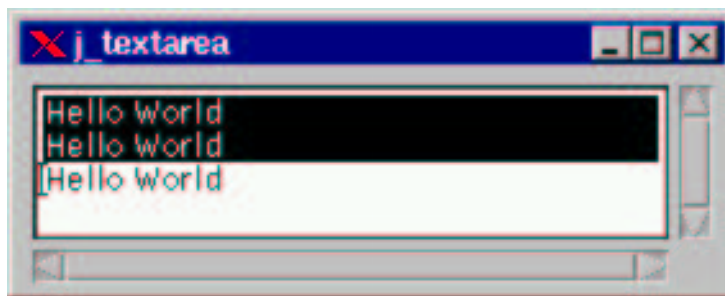
Arguments `obj integer`
 `columns integer`

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	procedure j_setcurpos (obj: integer; pos: integer) ;
Arguments	obj integer pos integer
Description	Change the location of the text cursor to the specified position pos .
Targets	Textarea, Textfield

setcursor

Synopsis	function j_setcursor (obj: integer; cursor: integer) return integer;
Arguments	obj integer cursor integer
Description	Changes the component 's obj cursor to the specified cursor .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setdanger

Synopsis procedure **j_setdanger** (obj: integer; val: integer) ;

Arguments obj integer
 val integer

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

Targets Meter

setdebug

Synopsis procedure **j_setdebug** (level: integer) ;

Arguments level integer

Description Sets the debuglevel to **level**.

setechochar

Synopsis	procedure j_setechochar (obj: integer; chr: character) ;
Arguments	obj integer chr character
Description	Changes the character chr that is used to echo all user input in the component .
Targets	Textfield

seteditable

Synopsis	procedure j_seteditable (obj: integer; bool: boolean) ;
Arguments	obj integer bool boolean
Description	Allows to make the component editable (bool=true) or read-only (bool=false).
Targets	Textarea, Textfield

setfixlayout

Synopsis	procedure j_setfixlayout (obj: integer) ;
Arguments	obj integer
Description	Adds a fixlayout manager to component obj (default layout manager).
Targets	Panel, Borderpanel, Window, Dialog, Frame

setflowfill

Synopsis	procedure j_setflowfill (obj: integer; bool: boolean) ;
Arguments	obj integer bool boolean
Description	Resizes all containing component to the height (width) of component obj . Needs a flowlayout manager.
Targets	Panel, Borderpanel, Window, Dialog, Frame

setflowlayout

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

setfocus

Synopsis	function j_setfocus (obj: integer) return integer;
Arguments	obj integer
Description	Directs the input focus to component obj .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setfontname

Synopsis procedure **j_setfontname** (obj: integer; name: integer) ;

Arguments obj integer
 name integer

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, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

Example

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

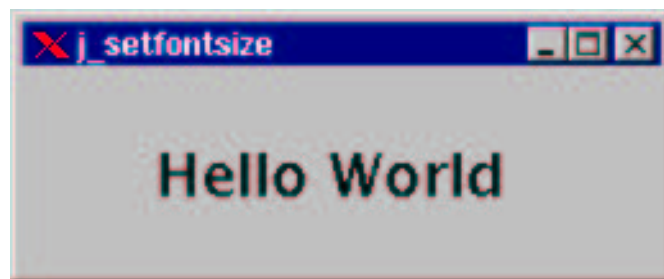


setfontsize

Synopsis	procedure j_setfontsize (obj: integer; size: integer) ;
Arguments	obj integer size integer
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, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

Example

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



setfontstyle

Synopsis `procedure j_setfontstyle (obj: integer; style: integer) ;`

Arguments `obj integer`
 `style integer`

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, Popupmenu

Example

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



setfont

Synopsis	procedure j_setfont (obj: integer; name: integer; style: integer; size: integer) ;								
Arguments	<table border="0"> <tr><td>obj</td><td>integer</td></tr> <tr><td>name</td><td>integer</td></tr> <tr><td>style</td><td>integer</td></tr> <tr><td>size</td><td>integer</td></tr> </table>	obj	integer	name	integer	style	integer	size	integer
obj	integer								
name	integer								
style	integer								
size	integer								
Description	Changes the font to the given characteristics name , style and size .								
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu								
Example	<pre> : label := j_label(jframe,'Hello World'); j_setfont(label,J_TIMES,J_PLAIN,18); : </pre>								

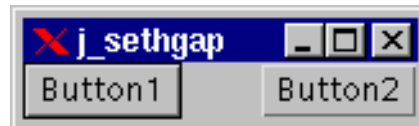


setgridlayout

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

sethgap

Synopsis	procedure j_sethgap (obj: integer; hgap: integer) ;
Arguments	obj integer hgap integer
Description	Sets the horizontal gap between components to hgap Pixel.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: j_flowlayout(frame,J_HORIZONTAL); button1 := j_button(frame,'Button1'); button2 := j_button(frame,'Button2'); j_sethgap(frame,30); :</pre>



seticon

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

setimage

Synopsis procedure **j_setimage** (obj: integer; image: integer) ;

Arguments obj integer
 image integer

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 procedure **j_setinsets** (obj: integer; top: integer; bottom: integer; left: integer; right: integer) ;

Arguments obj integer
 top integer
 bottom integer
 left integer
 right integer

Description Set the insets to the specified values.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

```

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

```



setmax

Synopsis	function j_setmax (obj: integer; val: integer) return integer;
Arguments	obj integer val integer
Description	Changes the maximum value for the component to val .
Targets	Scrollbar, Meter, Progress

setmin

Synopsis function **j_setmin** (obj: integer; val: integer) return integer;

Arguments obj integer
 val integer

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

Targets Scrollbar, Meter, Progress

setnamedcolorbg

Synopsis	procedure j_setnamedcolorbg (obj: integer; color: integer) ;
Arguments	obj integer color integer
Description	Sets the background color to a predefined color .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setnamedcolor

Synopsis procedure **j_setnamedcolor** (obj: integer; color: integer) ;

Arguments obj integer
 color integer

Description Sets the foreground color to a predefined **color**.

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

setnolayout

Synopsis	procedure j_setnolayout (obj: integer) ;
Arguments	obj integer
Description	Removes the current layout manager from component obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame

start

Synopsis procedure **j_start** (port: integer) ;

Arguments port integer

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( not j_start ) then begin
:
```

setpos

Synopsis	procedure j_setpos (obj: integer; xpos: integer; ypos: integer) ;
Arguments	obj integer xpos integer ypos integer
Description	Relocates the component obj to the specified Position (xpos,ypos).
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setradiogroup

Synopsis	function j_setradiogroup (rbutton,: integer; rgroup: integer) return integer;
Arguments	rbutton, integer rgroup integer
Description	Sets radiobuttons rbutton group to be the specified radiogroup rgroup . If the radiobuttons is already in a different radiogroup, it is first taken out of that group.
Targets	Radiobutton

setresizable

Synopsis procedure **j_setresizable** (obj: integer; resizable: boolean) ;

Arguments obj integer
 resizable boolean

Description The component cannot be resized, if **resizable** is false .

Targets Dialog, Frame

Example

```
:  
frame := j_frame('fixsized Frame');  
j_setresizable(frame,false);  
:
```

setrows

Synopsis procedure **j_setrows** (obj: integer; rows: integer) ;

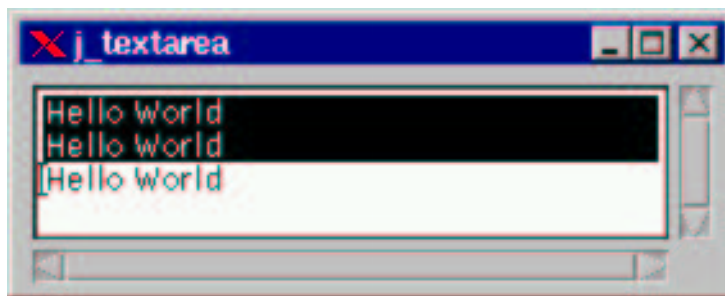
Arguments obj integer
 rows integer

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

Targets Textarea, Gridlayout

Example

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



setshortcut

Synopsis	procedure j_setshortcut (obj: integer; chr: character) ;
Arguments	obj integer chr character
Description	Changes the shortcut chr of the component .
Targets	MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

setsize

Synopsis procedure **j_setsize** (obj: integer; width: integer; height: integer) ;

Arguments obj integer
 width integer
 height integer

Description Resizes component **obj** to specified **width** and **height**.

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, 'Button');  
j_setsize(button, 100, 100);  
:
```



setslidesize

Synopsis	function j_setslidesize (obj: integer; val: integer) return integer;
Arguments	obj integer val integer
Description	Changes the slide size to val .
Targets	Scrollbar

setstate

Synopsis procedure **j_setstate** (obj: integer; bool: boolean) ;

Arguments obj integer
 bool boolean

Description The component becomes selected, if **bool** is true .

Targets Checkbox, Radiobutton, CheckmenuItem, Led

settext

Synopsis `procedure j_settext (obj: integer; str: string) ;`


Arguments `obj integer`
 `str string`

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: integer; val: integer) return integer;
Arguments	obj integer val integer
Description	Changes the unit increment amount for the component to val
Targets	Scrollbar

setvalue

Synopsis	procedure j_setvalue (obj: integer; val: integer) ;
Arguments	obj integer val integer
Description	Changes the current value of the component to val .
Targets	Scrollbar, Progress, Meter, Sevensegment

setvgap

Synopsis	procedure j_setvgap (obj: integer; vgap: integer) ;
Arguments	obj integer vgap integer
Description	Sets the vertical gap between components to hgap Pixel.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: j_setflowlayout(frame,J_VERTICAL); button1 := j_button(frame,'Button1'); button2 := j_button(frame,'Button2'); j_setvgap(frame,30); :</pre>



setxor

Synopsis	procedure j_setxor (obj: integer; bool: boolean) ;
Arguments	obj integer bool boolean
Description	Changes painting mode to XOR mode, if bool = 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

sevenssegment

Synopsis function **j_sevenssegment** (obj: integer; color: integer) return
integer;

Arguments obj integer
 color integer

Description Creates a new sevenssegment 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_sevenssegment(frame,J_GREEN);  
j_setvalue(seven,5);  
:
```



showpopup

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

show

Synopsis procedure **j_show** (obj: integer) ;

Arguments obj integer

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: integer) return integer;

Arguments msec integer

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

start

Synopsis function **j_start** return boolean;

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

Example

```
:
if( not j_start ) then begin
  writeln('cannot connect to JAPI Kernel');
  halt;
end;
:
```

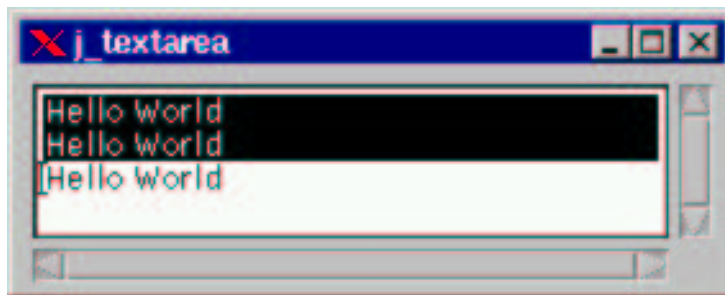

sync

Synopsis `procedure j_sync ;`

Description Synchronizes the application with the JAPI kernel.

textarea

Synopsis	function j_textarea (obj: integer; rows: integer; columns: integer) return integer;
Arguments	obj integer rows integer columns integer
Description	Creates a new textarea component with the specified number of rows columns and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: frame := j_frame('j_textarea'); text := j_textarea(frame,30,4) :</pre>



textfield

Synopsis function **j_textfield** (obj: integer; columns: integer) return
integer;

Arguments obj integer
 columns integer

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	procedure j_translate (obj: integer; x: integer; y: integer) ;
Arguments	obj integer x integer y integer
Description	Moves the origin of drawing operations to (x , y).
Targets	Canvas, Image, Printer

vscrollbar

Synopsis `function j_vscrollbar (obj: integer) return integer;`

Arguments `obj integer`

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: integer; kind: integer) return integer;
Arguments	window integer kind integer
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 : <ul style="list-style-type: none">• J_ACTIVATED : An event occurs when the component is activated.• J_DEACTIVATED : An event occurs when the component is deactivated.• J_OPENED : An event occurs when the component has been opened.• J_CLOSED : An event occurs when the component has been closed.• J_ICONFIED : An event occurs when the component is iconfied.• J_DEICONFIED : An event occurs when the component is deiconfied.• J_CLOSING : An event occurs when the close icon has been clicked .
Targets	Window, Dialog, Frame

window

Synopsis	function j_window (obj: integer) return integer;
Arguments	obj integer
Description	Creates a new simple window and returns its event number.
Targets	Frame
Example	<pre>: window := j_window(frame); label := j_label(window,'Mouse pressed at ... '); j_setnamedcolorbg(label,J_YELLOW); :</pre>

Mouse pressed at 108:179