

japi

Reference Manual

Merten Joost

Inhaltsverzeichnis

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

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

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

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

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

translate	296
vscrollbar	297
windowlistener	298
window	299

Teil I

Reference

Kapitel 1

Components

Button

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

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

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

Borderpanel

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

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

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

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

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

Canvas

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

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

j_fillrect	<i>procedure j_fillrect (integer obj , integer x , integer y , integer width , integer height)</i> Draws an filled rectangle from (x,y) of size width x height .
j_fillroundrect	<i>procedure j_fillroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)</i> Draws an filled rectangle from (x,y) of size width x height with rounded corners. arcx and arcy specify the radius of rectangle corners.
j_focuslistener	<i>integer function j_focuslistener (integer obj)</i> Adds a new focus listener to canvas obj , and returns its event number.
j_getfontascent	<i>integer function j_getfontascent (integer obj)</i> Returns the ascent (space above the baseline) of the actual font of canvas obj .
j_getfontheight	<i>integer function j_getfontheight (integer obj)</i> Returns the total pixel height of the actual font of canvas obj .
j_getheight	<i>integer function j_getheight (integer obj)</i> Returns the height of canvas obj .
j_getimage	<i>integer function j_getimage (integer obj)</i> Copy the contents of canvas obj into an image and return its eventnumber.
j_getimagesource	<i>integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)</i> Returns an image of the specified size (x, y, width, height) of canvas . The red, green and blue values of each pixel will be stored in r, g, b
j_getparentid	<i>integer function j_getparentid (integer obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getparent	<i>integer function j_getparent (integer obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j_getscaledimage	<i>integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)</i> Copy the contents of the rectangular area defined by x, y , width sw , and height sh into an image and return its eventnumber. The image will be scaled to target width th and target height th .
j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Returns the length of str of the actual font of canvas obj .
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Returns the width of canvas obj .
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Returns the current horizontal position of canvas obj in its parent's coordinate space.

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

j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Changes the font to the given style .
j_setnamedcolorbg	<i>procedure j_setnamedcolorbg (integer obj , integer color)</i> Sets the background color to a predefined color .
j_setnamedcolor	<i>procedure j_setnamedcolor (integer obj , integer color)</i> Sets the foreground color to a predefined color .
j_setpos	<i>procedure j_setpos (integer obj , integer xpos , integer ypos)</i> Relocates the canvas obj to the specified Position (xpos,ypos).
j_setsize	<i>procedure j_setsize (integer obj , integer width , integer height)</i> Resizes canvas obj to specified width and height .
j_setxor	<i>procedure j_setxor (integer obj , integer bool)</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 (integer obj)</i> Shows the canvas obj .
j_translate	<i>procedure j_translate (integer obj , integer x , integer y)</i> Moves the origin of drawing operations to (x, y).

Checkbox

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

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

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

Checkmenuitem

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

Choice

j.choice	<i>integer function j.choice (integer obj)</i> Creates a new choice component and returns its event number.
j.additem	<i>procedure j.additem (integer obj , character*(*) str)</i> adds a new item containing str to choice obj .
j.add	<i>procedure j.add (integer obj , integer cont)</i> Adds choice obj to container cont
j.componentlistener	<i>integer function j.componentlistener (integer obj , integer kind)</i> Adds a new componentlistener to choice obj , and returns its event number. An event occurs, if the user action is of kind kind .
j.disable	<i>procedure j.disable (integer obj)</i> Disables choice obj so that it is unresponsive to user interactions
j.dispose	<i>procedure j.dispose (integer obj)</i> Releases the resources of the choice obj .
j.enable	<i>procedure j.enable (integer obj)</i> enables the choice obj .
j.focuslistener	<i>integer function j.focuslistener (integer obj)</i> Adds a new focus listener to choice obj , and returns its event number.
j.getfontascent	<i>integer function j.getfontascent (integer obj)</i> Returns the ascent (space above the baseline) of the actual font of choice obj .
j.getfontheight	<i>integer function j.getfontheight (integer obj)</i> Returns the total pixel height of the actual font of choice obj .
j.getheight	<i>integer function j.getheight (integer obj)</i> Returns the height of choice obj .
j.getitemcount	<i>integer function j.getitemcount (integer obj)</i> Returns the number of items of choice obj .
j.getitem	<i>procedure j.getitem (integer obj , integer item , character*(*) str)</i> returns the label of the given item .
j.getparentid	<i>integer function j.getparentid (integer obj)</i> Returns the parent event number of component obj . If obj is a frame -1 will be returned.
j.getparent	<i>integer function j.getparent (integer obj)</i>

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

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

Dialog

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

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

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

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

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

- j_textfield** *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.
- j_vscrollbar** *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.
- j_windowlistener** *integer function j_windowlistener (integer window , integer kind)*
Adds a new windowlistener to **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.

Focuslistener

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

Frame

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

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

j_getstringwidth	<i>integer function j_getstringwidth (integer obj , character*(*) str)</i> Returns the length of str of the actual font of frame obj .
j_gettext	<i>procedure j_gettext (integer obj , character*(*) str)</i> returns the frame 's text or label.
j_getwidth	<i>integer function j_getwidth (integer obj)</i> Returns the width of frame obj .
j_getxpos	<i>integer function j_getxpos (integer obj)</i> Returns the current horizontal position of frame obj in its parent's coordinate space.
j_getypos	<i>integer function j_getypos (integer obj)</i> Returns the current vertical position of frame obj in its parent's coordinate space.
j_graphicbutton	<i>integer function j_graphicbutton (integer obj , character*(*) filename)</i> Creates a new graphicbutton component with the image loaded from filename and returns its event number.
j_graphiclabel	<i>integer function j_graphiclabel (integer obj , character*(*) str)</i> Creates a new graphiclabel component with the image loaded from filename and returns its event number.
j_hide	<i>procedure j_hide (integer obj)</i> Hides the frame obj .
j_hscrollbar	<i>integer function j_hscrollbar (integer obj)</i> Creates a new horizontal scrollbar and returns its event number.
j_isparent	<i>integer function j_isparent (integer obj , integer cont)</i> Returns .true. if cont is parent of obj , .false. otherwise.
j_isresizable	<i>integer function j_isresizable (integer obj)</i> returns true if frame is resizable, false otherwise
j_isvisible	<i>integer function j_isvisible (integer obj)</i> Returns .true. if obj is visible, .false. otherwise.
j_keylistener	<i>integer function j_keylistener (integer obj)</i> Adds a new key listener to frame obj , and returns its event number.
j_label	<i>integer function j_label (integer obj , character*(*) label)</i> Creates a new label component with the specified label and returns its event number.
j_led	<i>integer function j_led (integer obj , integer style , integer color)</i> Creates a new led component with the specified style and the specified color .
j_line	<i>integer function j_line (integer obj , integer orient , integer style , integer length)</i>

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

	Sets the alignment in frame obj to align . Needs a flowlayout Manager.
j_setborderlayout	<i>procedure j_setborderlayout (integer obj)</i> Adds a borderlayout manager to frame obj .
j_setborderpos	<i>procedure j_setborderpos (integer obj , integer pos)</i> Moves frame obj at a certain position. The outer container needs a border layout manager.
j_setcolorbg	<i>procedure j_setcolorbg (integer obj , integer r , integer g , integer b)</i> Sets the background color to the (r , g , b) values.
j_setcolor	<i>procedure j_setcolor (integer obj , integer r , integer g , integer b)</i> Sets the foreground color to the (r , g , b) values.
j_setcursor	<i>integer function j_setcursor (integer obj , integer cursor)</i> Changes the frame 's obj cursor to the specified cursor .
j_setfixlayout	<i>procedure j_setfixlayout (integer obj)</i> Adds a fixlayout manager to frame obj (default layout manager).
j_setflowfill	<i>procedure j_setflowfill (integer obj , integer bool)</i> Resizes all containing component to the height (width) of frame obj . Needs a flowlayout manager.
j_setflowlayout	<i>procedure j_setflowlayout (integer obj , integer align)</i> Adds a flowlayout manager to frame obj with the specified alignment .
j_setfocus	<i>integer function j_setfocus (integer obj)</i> Directs the input focus to frame obj .
j_setfontname	<i>procedure j_setfontname (integer obj , integer name)</i> Changes the font to the given name .
j_setfont	<i>procedure j_setfont (integer obj , integer name , integer style , integer size)</i> Changes the font to the given characteristics name , style and size .
j_setfontsize	<i>procedure j_setfontsize (integer obj , integer size)</i> Changes the font to the given size .
j_setfontstyle	<i>procedure j_setfontstyle (integer obj , integer style)</i> Changes the font to the given style .
j_setgridlayout	<i>procedure j_setgridlayout (integer obj , integer row , integer col)</i> Adds a gridlayout manager to frame obj with the specified rows and columns .
j_sethgap	<i>procedure j_sethgap (integer obj , integer hgap)</i> Sets the horizontal gap between components to hgap Pixel.
j_seticon	<i>procedure j_seticon (integer frame , integer icon)</i> Sets the image icon to display when the frame is iconized. Not all platforms support the concept of iconizing a window.

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

Helpmenu

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

Hscrollbar

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

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

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

Graphicbutton

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

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

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

Graphiclabel

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

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

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

Image

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

- Draws a series of line segments based on first **len** elements in **x** and **y**.
- j_drawrect** *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.
- j_drawroundrect** *procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j_drawscaledimage** *procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)*
Copy the contents of the rectangular area defined by **x**, **y**, width **sw**, and height **sh** of the **image** to position **(tx, ty)**. The area will be scaled to target width **th** and target height **th**.
- j_drawstring** *procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)*
Draws text on screen at position **(x,y)**.
- j_fillarc** *procedure j_fillarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
Draws an filled arc from angle **arc1** to angle **arc2** with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j_fillcircle** *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
Draws an filled circle with center **(x, y)** and radius **x**.
- j_filloval** *procedure j_filloval (integer obj , integer x , integer y , integer rx , integer ry)*
Draws an filled oval with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j_fillpolygon** *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
Draws an filled polygon based on first **len** elements in **x** and **y**.
- j_fillrect** *procedure j_fillrect (integer obj , integer x , integer y , integer width , integer height)*
Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j_fillroundrect** *procedure j_fillroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j_getheight** *integer function j_getheight (integer obj)*
Returns the height of image **obj**.
- j_getimage** *integer function j_getimage (integer obj)*
Copy the contents of image **obj** into an image and return its eventnumber.

- j_getimagesource** *integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Returns an image of the specified size (**x**, **y**, **width**, **height**) of image . The red, green and blue values of each pixel will be stored in **r**, **g**, **b**
- j_getscaledimage** *integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)*
Copy the contents of the rectangular area defined by **x**, **y**, width **sw**, and height **sh** into an image and return its eventnumber. The image will be scaled to target width **th** and target height **th**.
- j_getwidth** *integer function j_getwidth (integer obj)*
Returns the width of image **obj**.
- j_print** *procedure j_print (integer obj)*
prints the image .
- j_setxor** *procedure j_setxor (integer obj , integer bool)*
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 (integer obj , integer x , integer y)*
Moves the origin of drawing operations to (**x**, **y**).

KeyListener

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

Label

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

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

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

Led

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

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

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

List

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

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

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

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

j_setsize

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

j_show

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

Menu

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

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

j_setshortcut

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

j_settext

procedure j_settext (integer obj , character(*) str)*
Sets the content or the label of the menu **obj** to **str**.

Menuitem

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

Meter

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

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

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

Mouselistener

- j_mouselistener** *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to mouselistener **obj**, and returns its event number.
An event occurs, if the user action is of kind **kind**.
- j_dispose** *procedure j_dispose (integer obj)*
Releases the resources of the mouselistener **obj**.
- j_getmousebutton** *integer function j_getmousebutton (integer mouselistener)*
Returns the latest used mousebutton.
- j_getmousex** *integer function j_getmousex (integer mouselistener)*
Returns the current horizontal position of the mouse in its parent's coordinate space.
- j_getmousey** *integer function j_getmousey (integer mouselistener)*
Returns the current vertical position of the mouse in its parent's coordinate space.

Panel

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

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

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

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

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

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

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

j_showpopup

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

Printer

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

- j_drawrect** *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
 Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.
- j_drawroundrect** *procedure j_drawroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
 Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j_drawscaledimage** *procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)*
 Copy the contents of the rectangular area defined by **x**, **y**, width **sw**, and height **sh** of the **image** to position **(tx, ty)**. The area will be scaled to target width **th** and target height **th**.
- j_drawstring** *procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)*
 Draws text on screen at position **(x,y)**.
- j_fillarc** *procedure j_fillarc (integer obj , integer x , integer y , integer rx , integer ry , integer arc1 , integer arc2)*
 Draws an filled arc from angle **arc1** to angle **arc2** with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j_fillcircle** *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
 Draws an filled circle with center **(x, y)** and radius **x**.
- j_filloval** *procedure j_filloval (integer obj , integer x , integer y , integer rx , integer ry)*
 Draws an filled oval with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j_fillpolygon** *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
 Draws an filled polygon based on first **len** elements in **x** and **y**.
- j_fillrect** *procedure j_fillrect (integer obj , integer x , integer y , integer width , integer height)*
 Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j_fillroundrect** *procedure j_fillroundrect (integer obj , integer x , integer y , integer width , integer height , integer arcx , integer arcy)*
 Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j_print** *procedure j_print (integer obj)*
 prints the printer .
- j_setxor** *procedure j_setxor (integer obj , integer bool)*
 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 (integer obj , integer x , integer y)*
 Moves the origin of drawing operations to **(x, y)**.

Progressbar

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

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

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

Radiobutton

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

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

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

Sevensegment

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

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

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

<h2>Scrollpane</h2>

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

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

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

Textarea

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

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

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

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

Textfield

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

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

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

- j_setnamedcolor** *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.
- j_setpos** *procedure j_setpos (integer obj , integer xpos , integer ypos)*
Relocates the textfield **obj** to the specified Position (**xpos,ypos**).
- j_setsize** *procedure j_setsize (integer obj , integer width , integer height)*
Resizes textfield **obj** to specified **width** and **height**.
- j_settext** *procedure j_settext (integer obj , character*(*) str)*
Sets the content or the label of the textfield **obj** to **str**.
- j_show** *procedure j_show (integer obj)*
Shows the textfield **obj**.

Vscrollbar

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

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

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

Window

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

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

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

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

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

Kapitel 2

Functions

additem

Synopsis procedure **j_additem** (integer obj , character*(*) str)

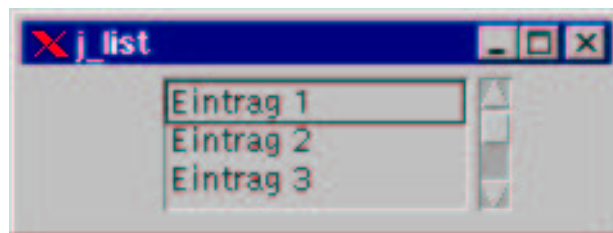
Arguments obj integer
 str character*(*)

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

Targets List, Choice

Example

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



add

Synopsis procedure **j_add** (integer obj , integer cont)

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** (integer obj , character*(*) title ,
 character*(*) text , character*(*) button)

Arguments

obj	integer
title	character*(*)
text	character*(*)
button	character*(*)

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

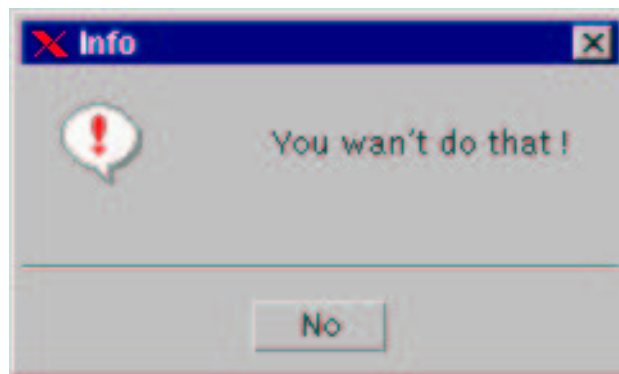
Targets Frame

Example

```

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

```



appendtext

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

Arguments obj integer
 text character*(*)

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

Targets Textarea

beep

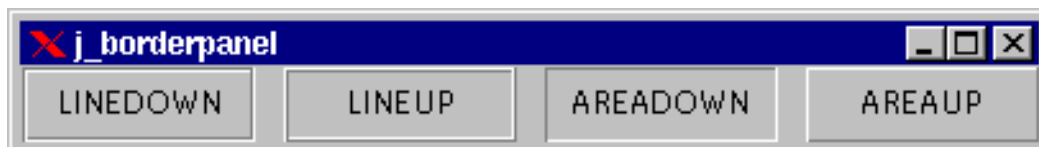
Synopsis procedure **j_beep** ()

Description Emits an audio beep.

borderpanel

Synopsis	integer function j_borderpanel (integer obj , integer type)
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	

```
:  
call 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	integer function j_button (integer obj , character*(*) label)
Arguments	obj integer label character*(*)
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 integer function **j_canvas** (integer obj , integer width , integer height)

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)  
call j_setnamedcolorbg(canvas,J_RED)  
:
```



checkbox

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

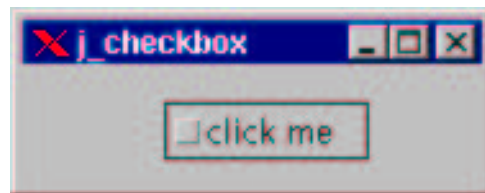
Arguments obj integer
label character*(*)

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

Arguments obj integer
 label character*(*)

Description creates a new checkmenuitem with the specified **label** and returns
 its event number.

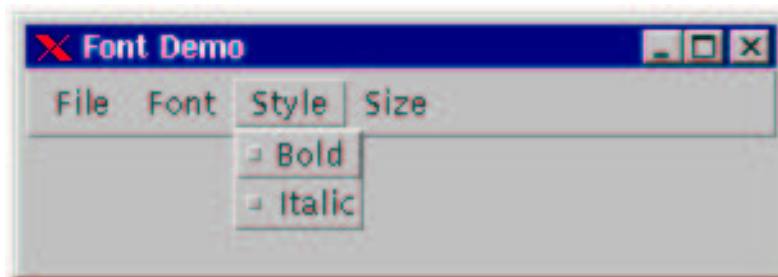
Targets Menu, Popupmenu, Helpmenu

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 (integer obj , character*(*) title ,
character*(*) text , character*(*) button1 , character*(*) button2
)`

Arguments

<code>obj</code>	<code>integer</code>
<code>title</code>	<code>character*(*)</code>
<code>text</code>	<code>character*(*)</code>
<code>button1</code>	<code>character*(*)</code>
<code>button2</code>	<code>character*(*)</code>

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.

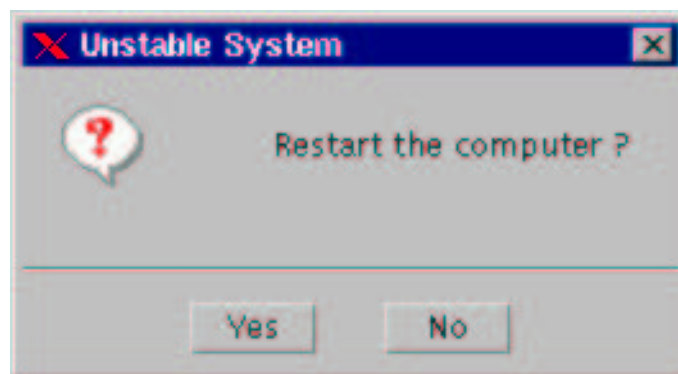
Targets `Frame`

Example

```

:
retval = j_checkbox2(frame,"Unstable System","Restart the computer ?",
                    "Yes","No")
:

```



checkbox3

Synopsis

```
procedure j_checkbox3 ( integer obj , character*(*) title ,
character*(*) text , character*(*) button1 , character*(*) button2
, character*(*) button3 )
```

Arguments

obj	integer
title	character*(*)
text	character*(*)
button1	character*(*)
button2	character*(*)
button3	character*(*)

Description

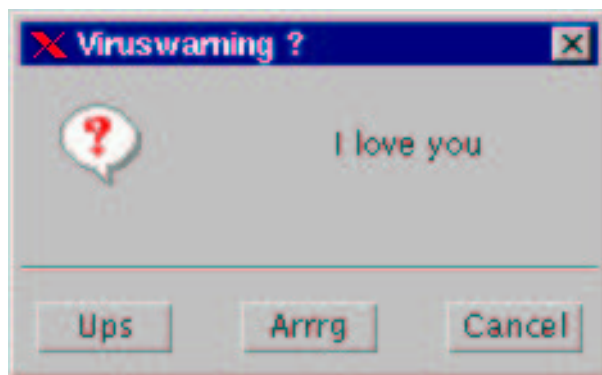
Shows a checkbox with the specified **title**, **text** and three buttons. Choiceboxes are modal dialogs, the application is blocked until a button or the closeicon is clicked. The focus is set to the first button. The return value is 0 if the closeicon is clicked, 1 for the first button, 2 for the second and 3 for the third one.

Targets

Frame

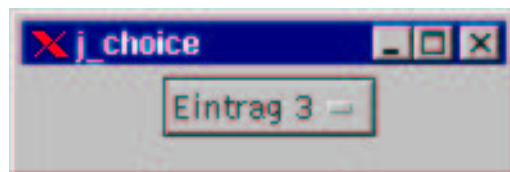
Example

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



choice

Synopsis	integer function j_choice (integer obj)
Arguments	obj integer
Description	Creates a new choice component and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: choice = j_choice(frame) call j_additem(choice,"Eintrag 1") call j_additem(choice,"Eintrag 2") :</pre>



cliprect

Synopsis	procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)
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	integer function j_componentlistener (integer obj , integer kind)
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 integer function **j_connect** (character*(*) hostname)

Arguments hostname character*(*)

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

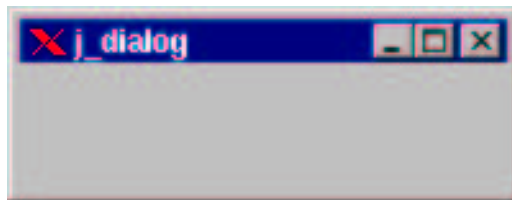
Synopsis	procedure j_delete (integer obj , integer start , integer end)
Arguments	obj integer start integer end integer
Description	Deletes text from starting position start to ending position end .
Targets	Textarea

deselect

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

dialog

Synopsis	integer function j_dialog (integer obj , character*(*) label)
Arguments	obj integer label character*(*)
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") call j_setsize(dialog,200,80) call j_show(dialog) :</pre>



disable

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

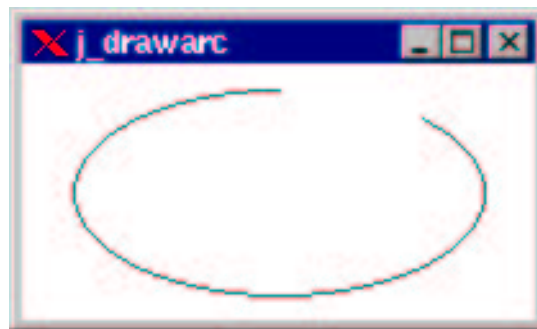
Arguments obj integer
 x integer
 y integer
 rx integer
 ry integer
 arc1 integer
 arc2 integer

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)  
call j_drawarc(canvas,100,50,80,40,45,-270)  
:
```



drawcircle

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

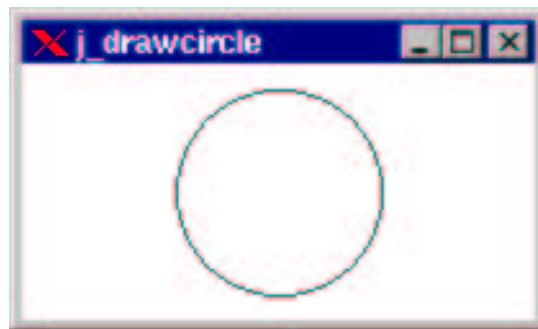
Arguments obj integer
 x integer
 y integer
 r integer

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

Targets Canvas, Image, Printer

Example

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



drawimagesource

Synopsis	procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)
Arguments	obj integer x integer y integer w integer h integer r array of integer g array of integer b array of integer
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 (integer obj , integer image , integer x , integer y)
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** (integer obj , integer x1 , integer y1 ,
integer x2 , integer y2)

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)  
call j_drawline(canvas,0,0,256,50)  
:
```



drawoval

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

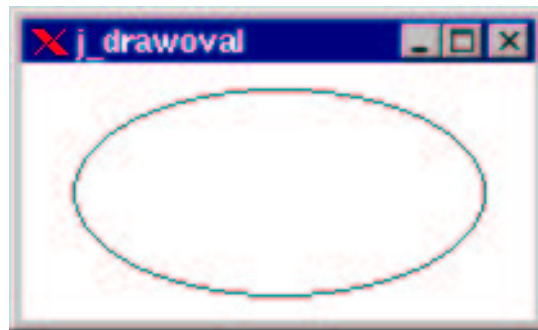
Arguments obj integer
 x integer
 y integer
 rx integer
 ry integer

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)  
call j_drawoval(canvas,100,50,80,40)  
:
```



drawpixel

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

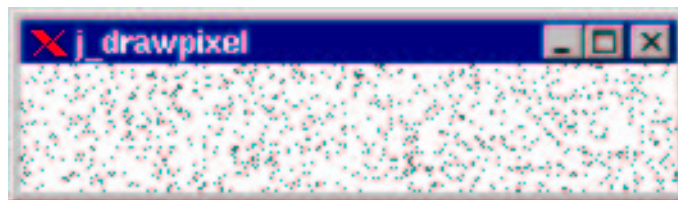
Arguments `obj integer`
 `x integer`
 `y integer`

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

Targets `Canvas, Image, Printer`

Example

```
:  
canvas = j_canvas(frame,256,50)  
do i=0,1000  
    call j_drawpixel(canvas, mod(j_random(),256), mod(j_random(),50))  
end do  
:
```



drawpolygon

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

Arguments

obj	integer
len	integer
x	array of integer
y	array of integer

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

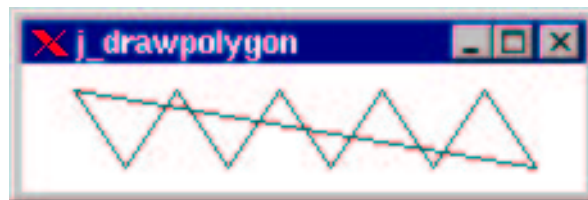
Targets Canvas, Image, Printer

Example

```

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

```



drawpolyline

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

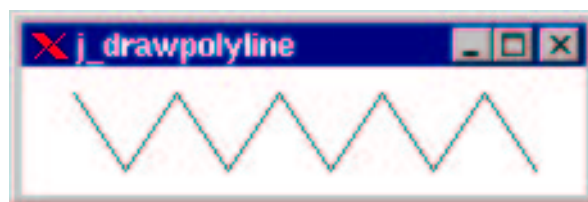
Arguments obj integer
 len integer
 x array of integer
 y array of integer

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

Targets Canvas, Image, Printer

Example

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



drawrect

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

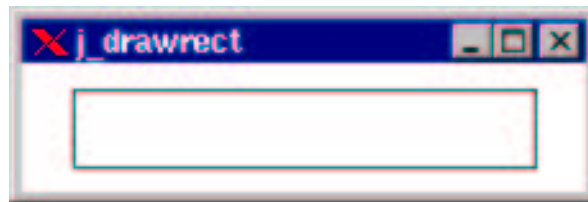
Arguments obj integer
 x integer
 y integer
 width integer
 height integer

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

Targets Canvas, Image, Printer

Example

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



drawroundrect

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

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.

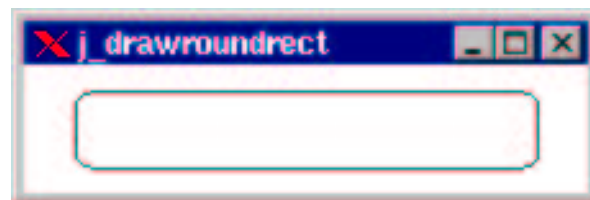
Targets Canvas, Image, Printer

Example

```

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

```



drawscaledimage

Synopsis	procedure j_drawscaledimage (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)
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** (integer obj , integer x , integer y ,
 character*(*) str)

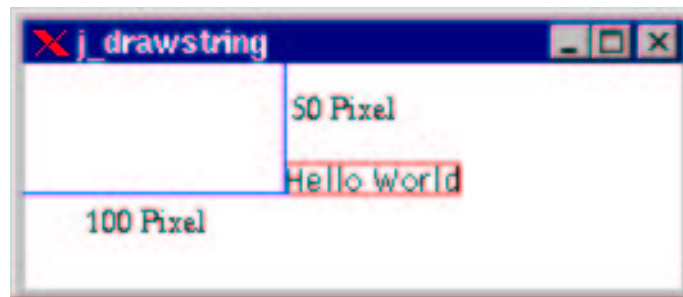
Arguments obj integer
 x integer
 y integer
 str character*(*)

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

Targets Canvas, Image, Printer

Example

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



enable

Synopsis	procedure j_enable (integer obj)
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 procedure **j_filedialog** (integer frame , character*(*) title ,
character*(*) directory , character*(*) filename)

Arguments

frame	integer
title	character*(*)
directory	character*(*)
filename	character*(*)

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.

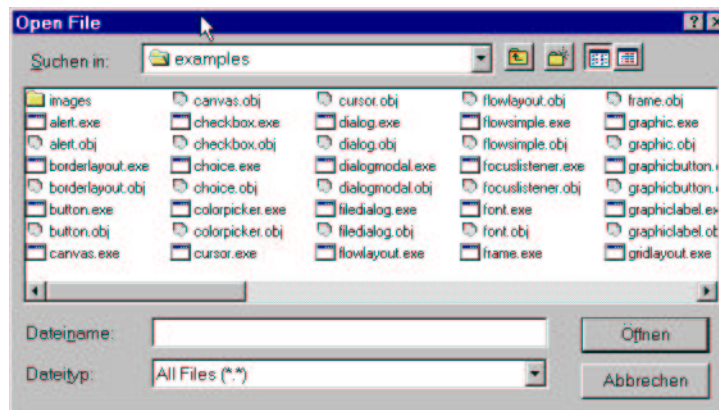
Targets Frame

Example

```

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

```



fileselect

Synopsis procedure **j_fileselect** (integer frame , character*(*) title ,
 character*(*) filter , character*(*) filename)

Arguments

frame	integer
title	character*(*)
filter	character*(*)
filename	character*(*)

Description Opens a fileslector 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 redi-
 rections via `j_connect()`;

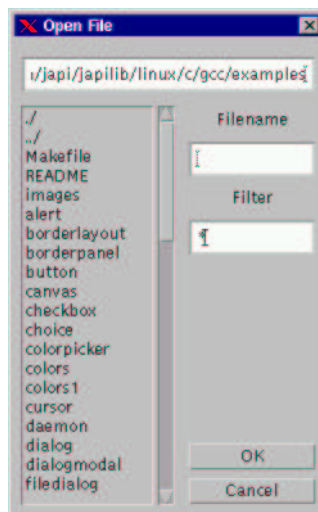
Targets Frame

Example

```

:
call j_fileselect(frame,"Open File","*",filename)
:

```



fillarc

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

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)
call j_fillarc(canvas,100,50,80,40,45,-270)
:
```



fillcircle

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

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)  
call j_fillcircle(canvas,100,50,40)  
:
```



filloval

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

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)  
call j_filloval(canvas,100,50,80,40)  
:
```



fillpolygon

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

Arguments obj integer
 len integer
 x array of integer
 y array of integer

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

Targets Canvas, Image, Printer

Example

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



fillrect

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

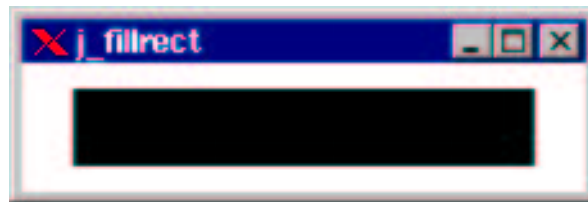
Arguments obj integer
 x integer
 y integer
 width integer
 height integer

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

Targets Canvas, Image, Printer

Example

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



fillroundrect

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

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.

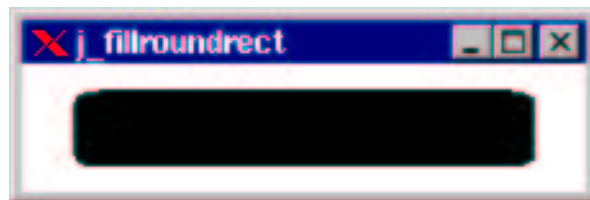
Targets Canvas, Image, Printer

Example

```

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

```



focuslistener

Synopsis	integer function j_focuslistener (integer obj)
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, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

frame

Synopsis integer function **j_frame** (character*(*) label)

Arguments label character*(*)

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

Example

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



getaction

Synopsis integer function **j_getaction** ()

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

getcolumns

Synopsis procedure **j_getcolumns** (integer obj)

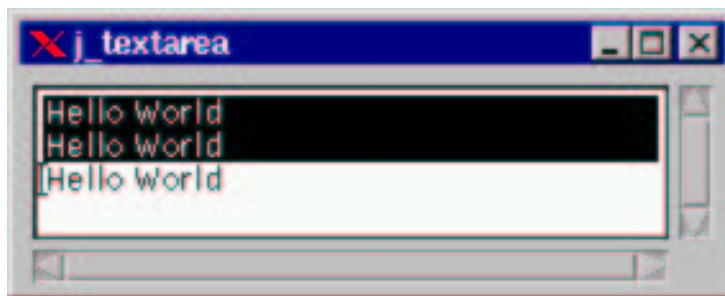
Arguments obj integer

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

Targets Textarea, Textfield, Gridlayout

Example

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



getcurpos

Synopsis	integer function j_getcurpos (integer obj)
Arguments	obj integer
Description	Returns the position, in characters, of the text cursor.
Targets	Textarea, Textfield

getdanger

Synopsis	procedure j_getdanger (integer obj)
Arguments	obj integer
Description	Returns the danger value of component obj .
Targets	Meter

getfontascent

Synopsis	integer function j_getfontascent (integer obj)
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	integer function j_getfontheight (integer obj)
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, Sevenssegment

getheight

Synopsis integer function **j_getheight** (integer obj)

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")
write(*,*) j_getheight(label)
:
> 22
```

getimagesource

Synopsis	integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)
Arguments	obj integer x integer y integer w integer h integer r array of integer g array of integer b array of integer
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	integer function j_getimage (integer obj)
Arguments	obj integer
Description	Copy the contents of component obj into an image and return its eventnumber.
Targets	Canvas, Image

getinheight

Synopsis integer function **j_getinheight** (integer cont)

Arguments cont integer

Description Returns the height of the client size.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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

getinsets

Synopsis	integer function j_getinsets (integer obj , integer side)
Arguments	obj integer side integer
Description	Returns the width of the specified inset. side can take the following values: <ul style="list-style-type: none"> • 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	<pre> : frame = j_frame("j_getinsets") write (*,*) j_getinsets(frame,J_TOP),j_getinsets(frame,J_BOTTOM), j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT) : > 25 5 5 6 </pre>



getinwidth

Synopsis integer function **j_getinwidth** (integer cont)

Arguments cont integer

Description Returns the width of the client size.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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

getitemcount

Synopsis integer function **j_getitemcount** (integer obj)

Arguments obj integer

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

Targets List, Choice

getitem

Synopsis	procedure j_getitem (integer obj , integer item , character*(*) str)
Arguments	obj integer item integer str character*(*)
Description	returns the label of the given item .
Targets	List, Choice

getkeychar

Synopsis integer function **j_getkeychar** (integer obj)

Arguments obj integer

Description Returns the ascii value of the last pressed key.

Targets Keylistener

getkeycode

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

getlayoutid

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

getlength

Synopsis	integer function j_getlength (integer obj)
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	integer function j_getmousebutton (integer mousetlistener)
Arguments	mousetlistenerinteger
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	Mousetlistener

getmousex

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

getmousey

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

getparentid

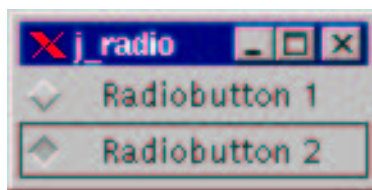
Synopsis	integer function j_getparentid (integer obj)
Arguments	obj integer
Description	Returns the parent event number of component obj . If obj is a frame -1 will be returned.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, 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 integer function **j_getparent** (integer obj)

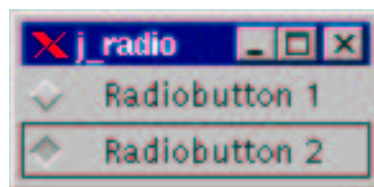
Arguments obj integer

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, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

Example

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



getrows

Synopsis procedure **j_getrows** (integer obj)

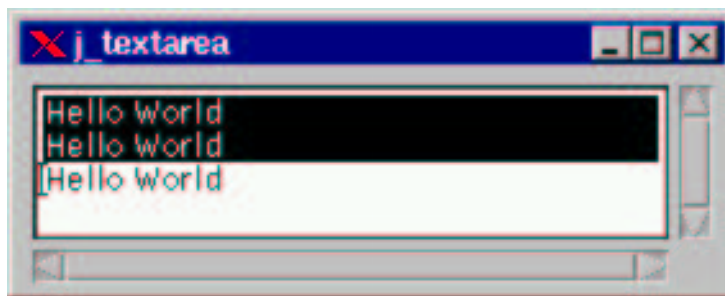
Arguments obj integer

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

Targets Textarea, Gridlayout

Example

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



getscaledimage

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

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

Example

```
:  
write(*,*) j_getscreenwidth(), j_getscreenheight()  
:  
  
> 1280 1024
```

getscreenwidth

Synopsis integer function **j_getscreenwidth** ()

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

Example

```
:  
write(*,*) j_getscreenwidth(), j_getscreenheight()  
:  
  
> 1280 1024
```


getselect

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

getselend

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

getselstart

Synopsis integer function **j_getselstart** (integer obj)

Arguments obj integer

Description Returns the initial position of any selected text.

Targets Textarea, Textfield

getseltext

Synopsis procedure **j_getseltext** (integer obj , character*(*) text)

Arguments obj integer
 text character*(*)

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

Targets Textarea, Textfield

getstate

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

getstringwidth

Synopsis	integer function j_getstringwidth (integer obj , character*(*) str)
Arguments	obj integer str character*(*)
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	procedure j_gettext (integer obj , character*(*) str)
Arguments	obj integer str character*(*)
Description	returns the component 's text or label.
Targets	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield
Example	<pre>character*256 str : label = j_label(frame,"Hello World") call j_gettext(label,str) write(*,*) str : > Hello World</pre>

getvalue

Synopsis integer function **j_getvalue** (integer obj)

Arguments obj integer

Description Returns the current setting of the scrollbar.

Targets Scrollbar

getviewportheight

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

getviewportwidth

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

getwidth

Synopsis	integer function j_getwidth (integer obj)
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	<pre>: label = j_getlabel(frame,"Hello World") write(*,*) j_getwidth(label) : > 84</pre>

getxpos

Synopsis	integer function j_getxpos (integer obj)
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	integer function j_getypos (integer obj)
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 integer function **j_graphicbutton** (integer obj , character*(*)
filename)

Arguments obj integer
 filename character*(*)

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 integer function **j_graphiclabel** (integer obj , character*(*)
str)

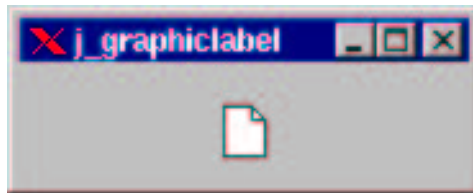
Arguments obj integer
 str character*(*)

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	integer function j_hasfocus (integer obj)
Arguments	obj integer
Description	Returns .true. if the component has the focus, .false. otherwise.
Targets	Focuslistener

helpmenu

Synopsis integer function **j_helpmenu** (integer obj , character*(*) label)

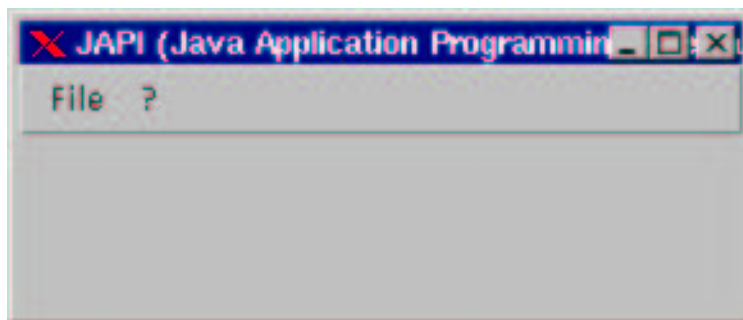
Arguments obj integer
label character*(*)

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,"?")
:
```

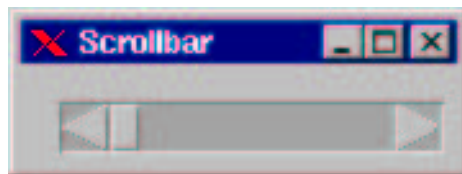


hide

Synopsis	procedure j_hide (integer obj)
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	integer function j_hscrollbar (integer obj)
Arguments	obj integer
Description	Creates a new horizontal scrollbar and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame, Scrollpane
Example	<pre>: scroll=j_hscrollbar(frame) call j_setpos(scroll,20,40) call j_setsize(scroll,150,20) :</pre>



image

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

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	integer function j_insert (integer obj , integer pos , character*(*) label)
Arguments	obj integer pos integer label character*(*)
Description	inserts a new item to component obj at position pos with the specified label .
Targets	List, Choice

inserttext

Synopsis	procedure j_inserttext (integer obj , character*(*) text , integer pos)
Arguments	obj integer text character*(*) pos integer
Description	Places additional text within the component at the given position pos .
Targets	Textarea

isparent

Synopsis	integer function j_isparent (integer obj , integer cont)
Arguments	obj integer cont integer
Description	Returns .true. if cont is parent of obj , .false. otherwise.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabeled, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup

isresizable

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

isselect

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

isvisible

Synopsis	integer function j_isvisible (integer obj)
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

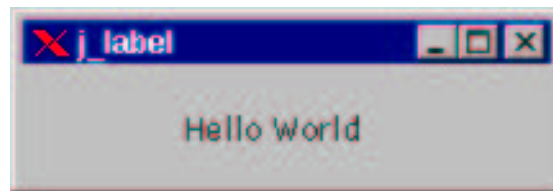
keylistener

Synopsis	integer function j_keylistener (integer obj)
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	integer function j_label (integer obj , character*(*) label)
Arguments	obj integer label character*(*)
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 integer function **j_led** (integer obj , integer style , integer color)

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)
:

```

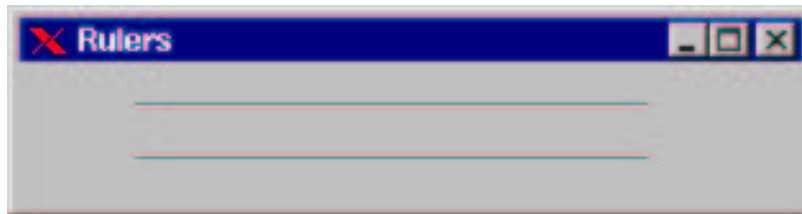


line

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

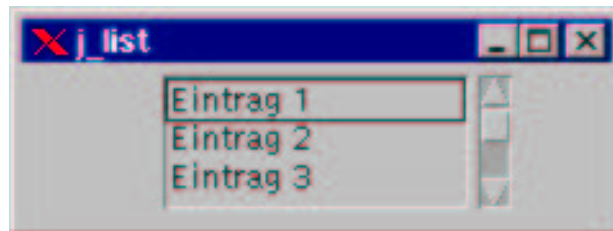
Example

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



list

Synopsis	integer function j_list (integer obj , integer rows)
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	<pre>: list = j_list(frame,3) call j_additem(list,"Eintrag 1") call j_additem(list,"Eintrag 2") :</pre>



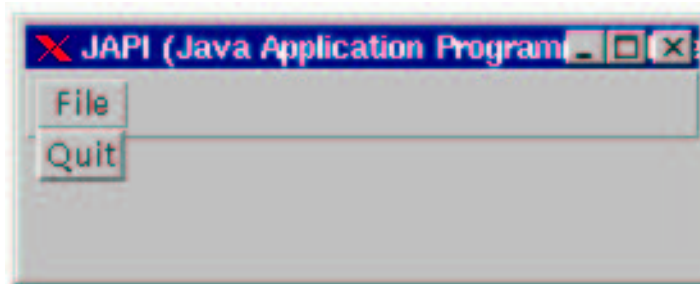
loadimage

Synopsis	integer function j_loadimage (character*(*) filename)
Arguments	filename character*(*)
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	integer function j_menubar (integer obj)
Arguments	obj integer
Description	Creates a new menubar and returns its event number.
Targets	Frame
Example	

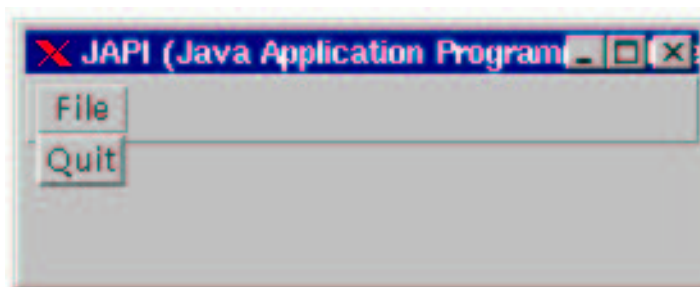
```
:  
frame  = j_frame("Menu Komponenten")  
menubar = j_menubar(frame)  
file   = j_menu(menubar,"File")  
quit   = j_menuitem(file,"Quit")  
:
```



menuitem

Synopsis	integer function j_menuitem (integer obj , character*(*) label)
Arguments	obj integer label character*(*)
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")
:
```

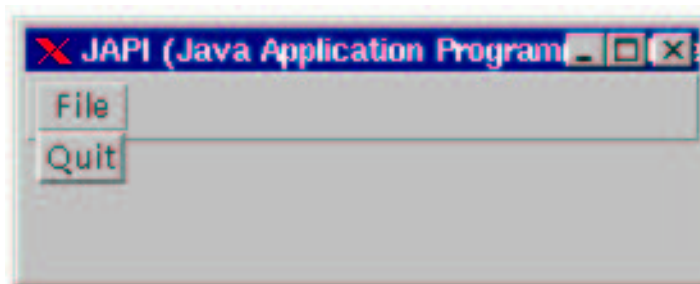


menu

Synopsis	integer function j_menu (integer obj , character*(*) str)
Arguments	obj integer str character*(*)
Description	Creates a new menu component with the specified label and returns its event number.
Targets	Menubar, Menu

Example

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



messagebox

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

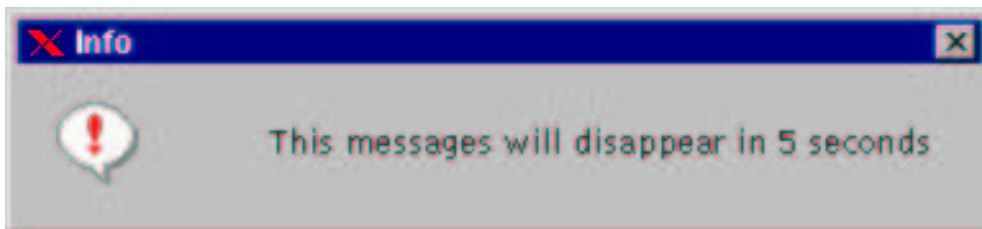
Arguments obj integer
 title character*(*)
 text character*(*)

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

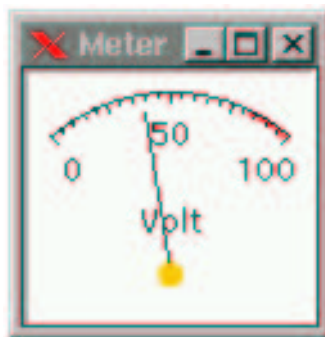
Example

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



meter

Synopsis	integer function j_meter (integer obj , character*(*) title)
Arguments	obj integer title character*(*)
Description	Creates a new pointer-instrument with the specified label titel and returns its event number. The meter has predifined values from 0 to 100. This can be canged via j_setmin() and j_setmax(). A danger value is set to 80 and can be justified with j_setdanger().
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre> : meter = j_meter(frame,"Volt") call j_setvalue(meter,40) : </pre>



mouselistener

Synopsis	integer function j_mouselistener (integer obj , integer kind)
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, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

multiplemode

Synopsis integer function **j_multiplemode** (integer obj , integer bool)

Arguments obj integer
 bool integer

Description if **bool** is .true. , selection mode is turned to multiplemode.

Targets List

nextaction

Synopsis integer function **j_nextaction** ()

Description Waits for the next event.

pack

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



panel

Synopsis integer function **j_panel** (integer obj)

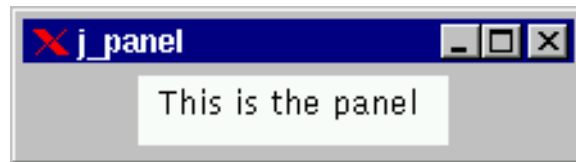
Arguments obj integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



popupmenu

Synopsis integer function **j_popupmenu** (integer obj , character*(*)
label)

Arguments obj integer
label character*(*)

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

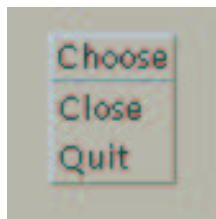
Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice,
Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window,
Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar,
Meter, Sensegment

Example

```

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

```



printer

Synopsis	integer function j_printer (integer frame)
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	<pre>: printer = j_printer(frame) call j_drawimage(printer,image,100,100) :</pre>

print

Synopsis	procedure j_print (integer obj)
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) : call j_print(frame) : </pre>



progressbar

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

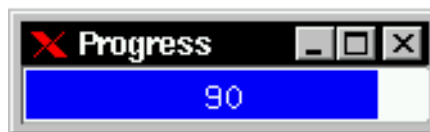
Arguments obj integer
 orient integer

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)  
call j_setvalue(progress,90)  
:
```



quit

Synopsis procedure **j_quit** ()

Description Cancels the connection to the JAPI Kernel.

radiobutton

Synopsis integer function **j_radiobutton** (integer obj , character*(*)
label)

Arguments obj integer
 label character*(*)

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	integer function j_radiogroup (integer obj)
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 integer function **j_random** ()

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

releaseall

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

release

Synopsis	procedure j_release (integer obj)
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 integer function **j_removeall** (integer obj)

Arguments obj integer

Description Removes all items from the component .

Targets List, Choice

removeitem

Synopsis	integer function j_removeitem (integer obj , character*(*) item)
Arguments	obj integer item character*(*)
Description	remove the first occurrence of item from the component .
Targets	List, Choice

remove

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

Arguments obj integer
 item integer

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

Targets List, Choice

replacetext

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

saveimage

Synopsis integer function **j_saveimage** (integer obj , character*(*) filename , integer filetype)

Arguments obj integer
 filename character*(*)
 filetype integer

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

- J_BMP Win32 Bitmap Format
- J_PPM Portable pixmap

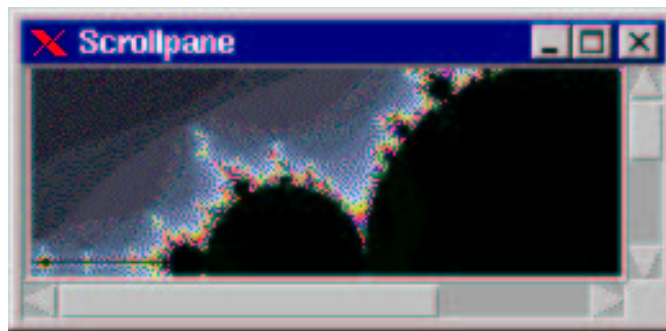
Example

```
:  
if(j_saveimage(canvas,"mandel.bmp",J_BMP) .eq. false)  
  write (*,*) "Error saving Bitmap file"  
:
```

scrollpane

Synopsis	integer function j_scrollpane (integer obj)
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")  
call j_setsize(scrollpane,240,100)  
:
```



selectall

Synopsis procedure **j_selectall** (integer obj)

Arguments obj integer

Description Selects all the text in the component .

Targets Textarea, Textfield

select

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

selecttext

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

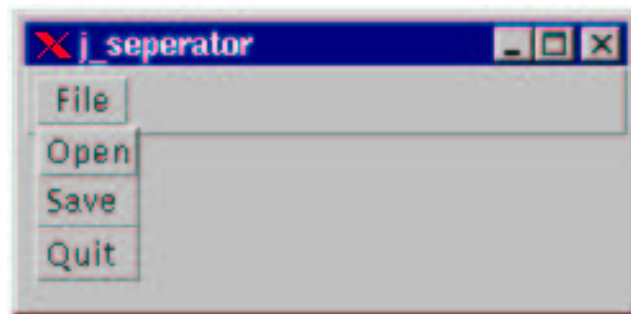
Arguments obj integer

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")
call j_seperator(file)
quit = j_menuitem(file,"Quit")
:
```



setalign

Synopsis	procedure j_setalign (integer obj , integer align)
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	integer function j_setblockinc (integer obj , integer val)
Arguments	obj integer val integer
Description	Changes the block increment amount for the component to val .
Targets	Scrollbar

setborderlayout

Synopsis	procedure j_setborderlayout (integer obj)
Arguments	obj integer
Description	Adds a borderlayout manager to component obj .
Targets	Panel, Borderpanel, Window, Dialog, Frame

setborderpos

Synopsis	procedure j_setborderpos (integer obj , integer pos)
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** (integer obj , integer r , integer g , ,
 integer b)

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")  
call j_setcolorbg(button,150,0,0)  
call j_settext(button,"Hello World")  
:
```



setcolor

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


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")  
call j_setcolor(button,150,0,0)  
call j_settext(button,"Hello World")  
:
```



setcolumns

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

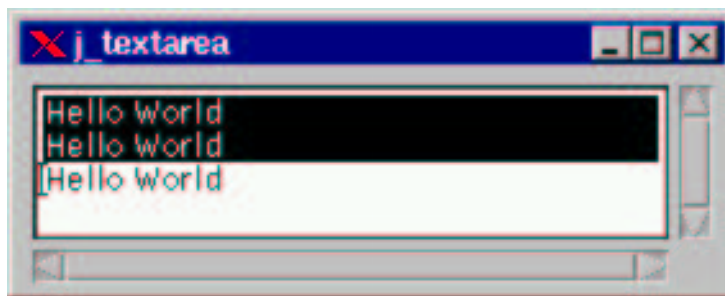
Arguments obj integer
 columns integer

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

Targets Textarea, Textfield, Gridlayout

Example

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



setcurpos

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

setcursor

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

setdanger

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

Arguments obj integer
 val integer

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

Targets Meter

setdebug

Synopsis procedure **j_setdebug** (integer level)

Arguments level integer

Description Sets the debuglevel to **level**.

setechochar

Synopsis	procedure j_setechochar (integer obj , character chr)
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 (integer obj , integer bool)
Arguments	obj integer bool integer
Description	Allows to make the component editable (bool =.true.) or read-only (bool =.false.).
Targets	Textarea, Textfield

setfixlayout

Synopsis	procedure j_setfixlayout (integer obj)
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 (integer obj , integer bool)
Arguments	obj integer bool integer
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 (integer obj , integer align)
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	integer function j_setfocus (integer obj)
Arguments	obj integer
Description	Directs the input focus to 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

setfontname

Synopsis procedure **j_setfontname** (integer obj , integer name)

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

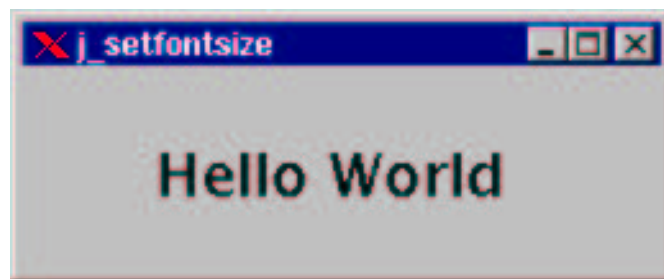
Example

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



setfontsize

Synopsis	procedure j_setfontsize (integer obj , integer size)
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	<pre>: label = j_label(jframe,"Hello World") call j_setfontsize(label,24) :</pre>



setfontstyle

Synopsis procedure **j_setfontstyle** (integer obj , integer style)

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")  
call j_setfontstyle(label,J_BOLD+J_ITALIC)  
:
```



setfont

Synopsis procedure **j_setfont** (integer obj , integer name , integer style
 , integer size)

Arguments

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, CheckBoxLayout, Menu, HelpMenu, Popupmenu

Example

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

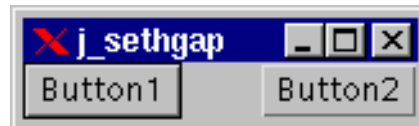


setgridlayout

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



seticon

Synopsis	procedure j_seticon (integer frame , integer icon)
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") call j_seticon(frame,j_loadimage("icon.gif")) :</pre>

setimage

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

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")  
call j_setimage(label,image)  
:
```



setinsets

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

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

```



setmax

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

setmin

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

Arguments obj integer
 val integer

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

Targets Scrollbar, Meter, Progress

setnamedcolorbg

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

start

Synopsis procedure **j_start** (integer port)

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

```
:  
call j_setport(12345)  
if( .not. j_start() ) then  
:
```

setpos

Synopsis	procedure j_setpos (integer obj , integer xpos , integer ypos)
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, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

setradiogroup

Synopsis	integer function j_setradiogroup (integer rbutton, , integer rgroup)
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** (integer obj , integer resizable)

Arguments obj integer
 resizable integer

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

Targets Dialog, Frame

Example

```
:  
frame = j_frame("fixsized Frame")  
call j_setresizable(frame, .false.)  
:
```

setrows

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

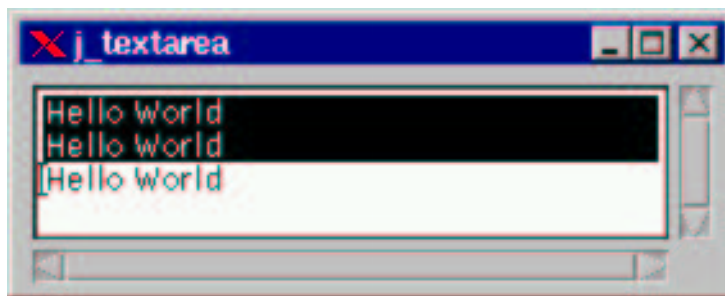
Arguments obj integer
 rows integer

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

Targets Textarea, Gridlayout

Example

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



setshortcut

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

setsize

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

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")  
call j_setsize(button,100,100)  
:
```



setslidesize

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

Arguments obj integer
 val integer

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

Targets Scrollbar

setstate

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

Arguments obj integer
 bool integer

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

Targets Checkbox, Radiobutton, Checkmenuitem, Led

settext

Synopsis procedure **j_settext** (integer obj , character*(*) str)


Arguments obj integer
 str character*(*)

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")  
call j_settext(button,"Goodbye")  
:
```



Goodbye

setunitinc

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

setvalue

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



setxor

Synopsis	procedure j_setxor (integer obj , integer bool)
Arguments	obj integer bool integer
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

sevensegment

Synopsis integer function **j_sevensegment** (integer obj , integer color)

Arguments obj integer
color integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



showpopup

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

show

Synopsis procedure **j_show** (integer obj)

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 integer function **j_sleep** (integer msec)

Arguments msec integer

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

start

Synopsis integer function **j_start** ()

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

Example

```
:  
if( .not. j_start() ) then  
    write(*,*) "can't connect to JAPI Kernel"  
    goto 20  
endif  
:
```


sync

Synopsis procedure **j_sync** ()

Description Synchronizes the application with the JAPI kernel.

textarea

Synopsis integer function **j_textarea** (integer obj , integer rows , integer columns)

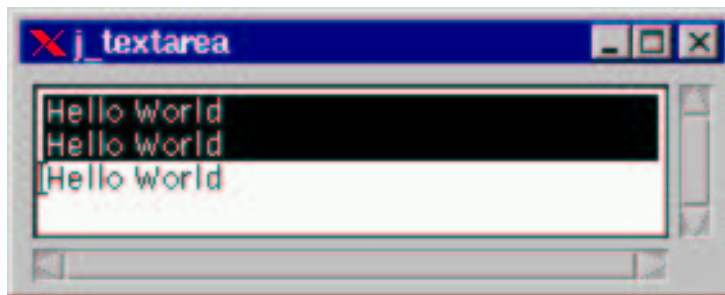
Arguments obj integer
 rows integer
 columns integer

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

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



textfield

Synopsis	integer function j_textfield (integer obj , integer columns)
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** (integer obj , integer x , integer y)

Arguments obj integer
 x integer
 y integer

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

Targets Canvas, Image, Printer

vscrollbar

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



windowlistener

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

window

Synopsis	integer function j_window (integer obj)
Arguments	obj integer
Description	Creates a new simple window and returns its event number.
Targets	Frame

Example

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

```
Mouse pressed at 108:179
```