

# japi

Reference Manual

Merten Joost



# Inhaltsverzeichnis

<b>I</b>	<b>Reference</b>	<b>5</b>
<b>1</b>	<b>Components</b>	<b>7</b>
	Button . . . . .	7
	Borderpanel . . . . .	10
	Canvas . . . . .	15
	Checkbox . . . . .	20
	CheckmenuItem . . . . .	23
	Choice . . . . .	24
	Dialog . . . . .	27
	Focuslistener . . . . .	33
	Frame . . . . .	34
	Helpmenu . . . . .	40
	Hscrollbar . . . . .	41
	Graphicbutton . . . . .	44
	Graphiclabel . . . . .	47
	Image . . . . .	50
	KeyListener . . . . .	53
	Label . . . . .	54
	Led . . . . .	57
	List . . . . .	60
	Menu . . . . .	64
	MenuItem . . . . .	66
	Meter . . . . .	67
	Mouselistener . . . . .	70
	Panel . . . . .	71
	PopupMenu . . . . .	76
	Printer . . . . .	78
	Progressbar . . . . .	81
	Radiobutton . . . . .	84
	Sevensegment . . . . .	87
	ScrollPane . . . . .	90
	Textarea . . . . .	93
	Textfield . . . . .	97
	Vscrollbar . . . . .	101
	Window . . . . .	104

<b>2 Functions</b>	<b>109</b>
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

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

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

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

Borderpanel
-------------

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

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

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



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

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

Canvas
--------

- j\_canvas**      *function j\_canvas ( obj: integer; width: integer; height: integer ) : integer;*  
 Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error  $-1$  will be returned.
- j\_add**      *procedure j\_add ( obj: integer; cont: integer ) ;*  
 Adds canvas **obj** to container **cont**
- j\_cliprect**      *procedure j\_cliprect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
 Changes current clipping region to the specified rectangle (**x**, **y**, **width**, **height**).
- j\_componentlistener**      *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
 Adds a new componentlistener to canvas **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**      *procedure j\_disable ( obj: integer ) ;*  
 Disables canvas **obj** so that it is unresponsive to user interactions
- j\_dispose**      *procedure j\_dispose ( obj: integer ) ;*  
 Releases the resources of the canvas **obj**.
- j\_drawarc**      *procedure j\_drawarc ( obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer ) ;*  
 Draws an unfilled arc from angle **arc1** to angle **arc2** with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j\_drawcircle**      *procedure j\_drawcircle ( obj: integer; x: integer; y: integer; r: integer ) ;*  
 Draws an unfilled circle with center (**x**, **y**) and radius **x**.
- j\_drawimage**      *procedure j\_drawimage ( obj: integer; image: integer; x: integer; y: integer ) ;*  
 Copies the image, given by its eventnumber **image**, to position (**x**, **y**).
- j\_drawimagesource**      *procedure j\_drawimagesource ( obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer ) ;*  
 Paints an image at Position (**x**, **y**) with **width** and **height**. The red, green and blue values of each pixel are given by the arrays **r**, **g**, **b**.
- j\_drawline**      *procedure j\_drawline ( obj: integer; x1: integer; y1: integer; x2: integer; y2: integer ) ;*  
 Draws a line connecting (**x1,y1**) and (**x2,y2**).
- j\_drawoval**      *procedure j\_drawoval ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;*

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

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

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

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

Checkbox
----------

- j\_checkbox**      *function j\_checkbox ( obj: integer; label: PChar ) : integer;*  
Creates a new checkbox component with the specified **label** and returns its event number.
- j\_add**            *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds checkbox **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to checkbox **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**        *procedure j\_disable ( obj: integer ) ;*  
Disables checkbox **obj** so that it is unresponsive to user interactions
- j\_dispose**       *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the checkbox **obj**.
- j\_enable**         *procedure j\_enable ( obj: integer ) ;*  
enables the checkbox **obj**.
- j\_focuslistener**   *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to checkbox **obj**, and returns its event number.
- j\_getfontascent**   *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of checkbox **obj**.
- j\_getfontheight**   *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of checkbox **obj**.
- j\_getheight**      *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of checkbox **obj**.
- j\_getparentid**     *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getparent**       *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getstate**        *function j\_getstate ( obj: integer ) : integer;*  
Returns true , if checkbox is selected, false otherwise.
- j\_getstringwidth**   *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of checkbox **obj**.



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

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

## CheckmenuItem

- j\_checkmenuItem**     *function j\_checkmenuItem ( obj: integer; label: PChar ) : integer;*  
creates a new checkmenuItem with the specified **label** and returns its event number.
- j\_disable**             *procedure j\_disable ( obj: integer ) ;*  
Disables checkmenuItem **obj** so that it is unresponsive to user interactions
- j\_dispose**             *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the checkmenuItem **obj**.
- j\_enable**                *procedure j\_enable ( obj: integer ) ;*  
enables the checkmenuItem **obj**.
- j\_getlength**            *function j\_getlength ( obj: integer ) : integer;*  
Returns the length of checkmenuItem 's label or text.
- j\_getstate**             *function j\_getstate ( obj: integer ) : integer;*  
Returns true , if checkmenuItem is selected, false otherwise.
- j\_gettext**              *function j\_gettext ( obj: integer; str: PChar ) : PChar;*  
returns the checkmenuItem 's text or label.
- j\_setfontname**         *procedure j\_setfontname ( obj: integer; name: integer ) ;*  
Changes the font to the given **name**.
- j\_setfont**               *procedure j\_setfont ( obj: integer; name: integer; style: integer; size: integer ) ;*  
Changes the font to the given characteristics **name**, **style** and **size**.
- j\_setfontsize**         *procedure j\_setfontsize ( obj: integer; size: integer ) ;*  
Changes the font to the given **size**.
- j\_setfontstyle**         *procedure j\_setfontstyle ( obj: integer; style: integer ) ;*  
Changes the font to the given **style**.
- j\_setshortcut**         *procedure j\_setshortcut ( obj: integer; chr: char ) ;*  
Changes the shortcut **chr** of the checkmenuItem .
- j\_setstate**             *procedure j\_setstate ( obj: integer; bool: boolean ) ;*  
The checkmenuItem becomes selected, if **bool** is true .
- j\_settext**              *procedure j\_settext ( obj: integer; str: PChar ) ;*  
Sets the content or the label of the checkmenuItem **obj** to **str**.

Choice
--------

- j\_choice**      *function j\_choice ( obj: integer ) : integer;*  
Creates a new choice component and returns its event number.
- j\_additem**      *procedure j\_additem ( obj: integer; str: PChar );*  
adds a new item containing **str** to choice **obj**.
- j\_add**      *procedure j\_add ( obj: integer; cont: integer );*  
Adds choice **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
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**      *procedure j\_disable ( obj: integer );*  
Disables choice **obj** so that it is unresponsive to user interactions
- j\_dispose**      *procedure j\_dispose ( obj: integer );*  
Releases the resources of the choice **obj**.
- j\_enable**      *procedure j\_enable ( obj: integer );*  
enables the choice **obj**.
- j\_focuslistener**      *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to choice **obj**, and returns its event number.
- j\_getfontascent**      *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of choice **obj**.
- j\_getfontheight**      *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of choice **obj**.
- j\_getheight**      *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of choice **obj**.
- j\_getitemcount**      *function j\_getitemcount ( obj: integer ) : integer;*  
Returns the number of items of choice **obj**.
- j\_getitem**      *function j\_getitem ( obj: integer; item: integer; str: PChar ) : PChar;*  
returns the label of the given **item**.
- j\_getparentid**      *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame  $-1$  will be returned.
- j\_getparent**      *function j\_getparent ( obj: integer ) : integer;*

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

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

Dialog
--------

- j\_dialog**      *function j\_dialog ( obj: integer; label: PChar ) : integer;*  
Creates a new dialog window with the specified **label** and returns its event number.
- j\_add**      *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds dialog **obj** to container **cont**
- j\_borderpanel**      *function j\_borderpanel ( obj: integer; type: integer ) : integer;*  
Creates a new borderpanel component with the style **type** and returns its event number.
- j\_button**      *function j\_button ( obj: integer; label: PChar ) : integer;*  
Creates a new button component with the specified **label** and returns its event number.
- j\_canvas**      *function j\_canvas ( obj: integer; width: integer; height: integer ) : integer;*  
Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error  $-1$  will be returned.
- j\_checkbox**      *function j\_checkbox ( obj: integer; label: PChar ) : integer;*  
Creates a new checkbox component with the specified **label** and returns its event number.
- j\_choice**      *function j\_choice ( obj: integer ) : integer;*  
Creates a new choice component and returns its event number.
- j\_componentlistener**      *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to dialog **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**      *procedure j\_disable ( obj: integer ) ;*  
Disables dialog **obj** so that it is unresponsive to user interactions
- j\_dispose**      *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the dialog **obj**.
- j\_enable**      *procedure j\_enable ( obj: integer ) ;*  
enables the dialog **obj**.
- j\_focuslistener**      *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to dialog **obj**, and returns its event number.
- j\_getfontascent**      *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of dialog **obj**.
- j\_getfontheight**      *function j\_getfontheight ( obj: integer ) : integer;*

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



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

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

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

- j\_textfield**      *function j\_textfield ( obj: integer; columns: integer ) : integer;*  
Creates a new textfield component with the specified number of **columns** and returns its event number.
- j\_vscrollbar**      *function j\_vscrollbar ( obj: integer ) : integer;*  
Creates a new vertical scrollbar and returns its event number.
- j\_windowlistener**      *function j\_windowlistener ( window: integer; kind: integer ) : integer;*  
Adds a new windowlistener to **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.

Focuslistener
---------------

- j\_focuslistener**      *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to focuslistener **obj**, and returns its event number.
- j\_dispose**            *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the focuslistener **obj**.
- j\_hasfocus**          *function j\_hasfocus ( obj: integer ) : integer;*  
Returns true if the focuslistener has the focus, false otherwise.

Frame
-------

- j\_frame**      *function j\_frame ( label: PChar ) : integer;*  
Creates a new frame component with the specified **label** and returns its event number.
- j\_add**      *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds frame **obj** to container **cont**
- j\_alertbox**      *procedure j\_alertbox ( obj: integer; title: PChar; text: PChar; button: PChar ) ;*  
Shows a alertbox with the specified **title**, **text** and **button**.
- j\_borderpanel**      *function j\_borderpanel ( obj: integer; type: integer ) : integer;*  
Creates a new borderpanel component with the style **type** and returns its event number.
- j\_button**      *function j\_button ( obj: integer; label: PChar ) : integer;*  
Creates a new button component with the specified **label** and returns its event number.
- j\_canvas**      *function j\_canvas ( obj: integer; width: integer; height: integer ) : integer;*  
Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error  $-1$  will be returned.
- j\_checkbox**      *function j\_checkbox ( obj: integer; label: PChar ) : integer;*  
Creates a new checkbox component with the specified **label** and returns its event number.
- j\_choicebox2**      *procedure j\_choicebox2 ( obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar ) ;*  
Shows a choicebox with the specified **title**, **text** and two buttons.
- j\_choicebox3**      *procedure j\_choicebox3 ( obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar; button3: PChar ) ;*  
Shows a choicebox with the specified **title**, **text** and three buttons.
- j\_choice**      *function j\_choice ( obj: integer ) : integer;*  
Creates a new choice component and returns its event number.
- j\_componentlistener**      *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to frame **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_dialog**      *function j\_dialog ( obj: integer; label: PChar ) : integer;*  
Creates a new dialog window with the specified **label** and returns its event number.
- j\_disable**      *procedure j\_disable ( obj: integer ) ;*

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

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

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



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

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

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

Helpmenu
----------

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

Hscrollbar
------------

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

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

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

Graphicbutton
---------------

- j\_graphicbutton**     *function j\_graphicbutton ( obj: integer; filename: PChar ) : integer;*  
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.
- j\_add**     *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds graphicbutton **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to graphicbutton **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**     *procedure j\_disable ( obj: integer ) ;*  
Disables graphicbutton **obj** so that it is unresponsive to user interactions
- j\_dispose**     *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the graphicbutton **obj**.
- j\_enable**     *procedure j\_enable ( obj: integer ) ;*  
enables the graphicbutton **obj**.
- j\_focuslistener**     *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to graphicbutton **obj**, and returns its event number.
- j\_getfontascent**     *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of graphicbutton **obj**.
- j\_getfontheight**     *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of graphicbutton **obj**.
- j\_getheight**     *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of graphicbutton **obj**.
- j\_getparentid**     *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getparent**     *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getstringwidth**     *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of graphicbutton **obj**.
- j\_getwidth**     *function j\_getwidth ( obj: integer ) : integer;*  
Returns the width of graphicbutton **obj**.



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

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

Graphiclabel
--------------

- j\_graphiclabel**      *function j\_graphiclabel ( obj: integer; str: PChar ) : integer;*  
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.
- j\_add**                *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds graphiclabel **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to graphiclabel **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**            *procedure j\_disable ( obj: integer ) ;*  
Disables graphiclabel **obj** so that it is unresponsive to user interactions
- j\_dispose**           *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the graphiclabel **obj**.
- j\_enable**             *procedure j\_enable ( obj: integer ) ;*  
enables the graphiclabel **obj**.
- j\_focuslistener**    *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to graphiclabel **obj**, and returns its event number.
- j\_getfontascent**    *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of graphiclabel **obj**.
- j\_getfontheight**    *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of graphiclabel **obj**.
- j\_getheight**         *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of graphiclabel **obj**.
- j\_getparentid**      *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getparent**         *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getstringwidth**   *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of graphiclabel **obj**.
- j\_getwidth**          *function j\_getwidth ( obj: integer ) : integer;*  
Returns the width of graphiclabel **obj**.

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

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

Image
-------

- j\_image**      *function j\_image ( width: integer; height: integer ) : integer;*  
Creates a new (memory) image component with the given **width** and **height** and returns its event number.
- j\_cliprect**      *procedure j\_cliprect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
Changes current clipping region to the specified rectangle (**x**, **y**, **width**, **height**).
- j\_dispose**      *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the image **obj**.
- j\_drawarc**      *procedure j\_drawarc ( obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer ) ;*  
Draws an unfilled arc from angle **arc1** to angle **arc2** with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j\_drawcircle**      *procedure j\_drawcircle ( obj: integer; x: integer; y: integer; r: integer ) ;*  
Draws an unfilled circle with center (**x**, **y**) and radius **x**.
- j\_drawimage**      *procedure j\_drawimage ( obj: integer; image: integer; x: integer; y: integer ) ;*  
Copies the image, given by its eventnumber **image**, to position (**x**, **y**).
- j\_drawimagesource**      *procedure j\_drawimagesource ( obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer ) ;*  
Paints an image at Position (**x**, **y**) with **width** and **height**. The red, green and blue values of each pixel are given by the arrays **r**, **g**, **b**.
- j\_drawline**      *procedure j\_drawline ( obj: integer; x1: integer; y1: integer; x2: integer; y2: integer ) ;*  
Draws a line connecting (**x1,y1**) and (**x2,y2**).
- j\_drawoval**      *procedure j\_drawoval ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;*  
Draws an unfilled oval with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j\_drawpixel**      *procedure j\_drawpixel ( obj: integer; x: integer; y: integer ) ;*  
Draws a pixel at (**x,y**).
- j\_drawpolygon**      *procedure j\_drawpolygon ( obj: integer; len: integer; x: var integer; y: var integer ) ;*  
Draws an unfilled polygon based on first **len** elements in **x** and **y**.
- j\_drawpolyline**      *procedure j\_drawpolyline ( obj: integer; len: integer; x: var integer; y: var integer ) ;*

- Draws a series of line segments based on first **len** elements in **x** and **y**.
- j\_drawrect**      *procedure j\_drawrect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.
- j\_drawroundrect**      *procedure j\_drawroundrect ( obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer ) ;*  
Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j\_drawscaledimage**      *procedure j\_drawscaledimage ( obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer ) ;*  
Copy the contents of the rectangular area defined by **x**, **y**, width **sw**, and height **sh** of the **image** to position **(tx, ty)**. The area will be scaled to target width **th** and target height **th**.
- j\_drawstring**      *procedure j\_drawstring ( obj: integer; x: integer; y: integer; str: PChar ) ;*  
Draws text on screen at position **(x,y)**.
- j\_fillarc**      *procedure j\_fillarc ( obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer ) ;*  
Draws an filled arc from angle **arc1** to angle **arc2** with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j\_fillcircle**      *procedure j\_fillcircle ( obj: integer; x: integer; y: integer; r: integer ) ;*  
Draws an filled circle with center **(x, y)** and radius **x**.
- j\_filloval**      *procedure j\_filloval ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;*  
Draws an filled oval with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j\_fillpolygon**      *procedure j\_fillpolygon ( obj: integer; len: integer; x: var integer; y: var integer ) ;*  
Draws an filled polygon based on first **len** elements in **x** and **y**.
- j\_fillrect**      *procedure j\_fillrect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j\_fillroundrect**      *procedure j\_fillroundrect ( obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer ) ;*  
Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j\_getheight**      *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of image **obj**.
- j\_getimage**      *function j\_getimage ( obj: integer ) : integer;*  
Copy the contents of image **obj** into an image and return its eventnumber.

- j\_getimagesource**     *function j\_getimagesource ( obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer ) : integer;*  
Returns an image of the specified size (**x**, **y**, **width**, **height**) of image . The red, green and blue values of each pixel will be stored in **r**, **g**, **b**
- j\_getscaledimage**     *function j\_getscaledimage ( obj: integer; x: integer; y: integer; sw: integer; sh: integer; tw: integer; th: integer ) : integer;*  
Copy the contents of the rectangular area defined by **x**, **y**, width **sw**, and height **sh** into an image and return its eventnumber. The image will be scaled to target width **th** and target height **th**.
- j\_getwidth**             *function j\_getwidth ( obj: integer ) : integer;*  
Returns the width of image **obj**.
- j\_print**                 *procedure j\_print ( obj: integer ) ;*  
prints the image .
- j\_setxor**                *procedure j\_setxor ( obj: integer; bool: boolean ) ;*  
Changes painting mode to XOR mode, if **bool** = true . In this mode, drawing the same object in the same color at the same location twice has no net effect.
- j\_translate**            *procedure j\_translate ( obj: integer; x: integer; y: integer ) ;*  
Moves the origin of drawing operations to (**x**, **y**).



## KeyListener

- j\_keylistener**      *function j\_keylistener ( obj: integer ) : integer;*  
Adds a new key listener to keylistener **obj**, and returns its event number.
- j\_dispose**          *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the keylistener **obj**.
- j\_getkeychar**      *function j\_getkeychar ( obj: integer ) : integer;*  
Returns the ascii value of the last pressed key.
- j\_getkeycode**      *function j\_getkeycode ( obj: integer ) : integer;*  
Returns the integer key code of the last pressed key.

Label
-------

- j\_label**      *function j\_label ( obj: integer; label: PChar ) : integer;*  
Creates a new label component with the specified **label** and returns its event number.
- j\_add**      *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds label **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to label **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**      *procedure j\_disable ( obj: integer ) ;*  
Disables label **obj** so that it is unresponsive to user interactions
- j\_dispose**      *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the label **obj**.
- j\_enable**      *procedure j\_enable ( obj: integer ) ;*  
enables the label **obj**.
- j\_focuslistener**      *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to label **obj**, and returns its event number.
- j\_getfontascent**      *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of label **obj**.
- j\_getfontheight**      *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of label **obj**.
- j\_getheight**      *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of label **obj**.
- j\_getparentid**      *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getparent**      *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getstringwidth**      *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of label **obj**.
- j\_gettext**      *function j\_gettext ( obj: integer; str: PChar ) : PChar;*  
returns the label 's text or label.
- j\_getwidth**      *function j\_getwidth ( obj: integer ) : integer;*

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

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

Led
-----

- j\_led**                    *function j\_led ( obj: integer; style: integer; color: integer ) : integer;*  
Creates a new led component with the specified **style** and the specified color **color**.
- j\_add**                    *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds led **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to led **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**                *procedure j\_disable ( obj: integer ) ;*  
Disables led **obj** so that it is unresponsive to user interactions
- j\_dispose**               *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the led **obj**.
- j\_enable**                *procedure j\_enable ( obj: integer ) ;*  
enables the led **obj**.
- j\_focuslistener**        *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to led **obj**, and returns its event number.
- j\_getfontascent**        *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of led **obj**.
- j\_getfontheight**        *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of led **obj**.
- j\_getheight**            *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of led **obj**.
- j\_getparentid**         *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getparent**            *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getstate**             *function j\_getstate ( obj: integer ) : integer;*  
Returns true , if led is selected, false otherwise.
- j\_getstringwidth**      *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of led **obj**.
- j\_getwidth**             *function j\_getwidth ( obj: integer ) : integer;*

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

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

List
------

- j\_list**                    *function j\_list ( obj: integer; rows: integer ) : integer;*  
Creates a new list component with the specified number of **rows** and returns its event number.
- j\_additem**                *procedure j\_additem ( obj: integer; str: PChar ) ;*  
adds a new item containing **str** to list **obj**.
- j\_add**                     *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds list **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to list **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_deselect**               *function j\_deselect ( obj: integer; item: integer ) : integer;*  
Deselects the item at the designated position **item**, if selected.
- j\_disable**                *procedure j\_disable ( obj: integer ) ;*  
Disables list **obj** so that it is unresponsive to user interactions
- j\_dispose**               *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the list **obj**.
- j\_enable**                 *procedure j\_enable ( obj: integer ) ;*  
enables the list **obj**.
- j\_focuslistener**        *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to list **obj**, and returns its event number.
- j\_getfontascent**        *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of list **obj**.
- j\_getfontheight**        *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of list **obj**.
- j\_getheight**             *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of list **obj**.
- j\_getitemcount**        *function j\_getitemcount ( obj: integer ) : integer;*  
Returns the number of items of list **obj**.
- j\_getitem**                *function j\_getitem ( obj: integer; item: integer; str: PChar ) : PChar;*  
returns the label of the given **item**.
- j\_getparentid**         *function j\_getparentid ( obj: integer ) : integer;*



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

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

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

**j\_setsize**

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

**j\_show**

*procedure j\_show ( obj: integer ) ;*  
Shows the list **obj**.

Menu
------

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

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

**j\_setshortcut**

*procedure j\_setshortcut ( obj: integer; chr: char ) ;*  
Changes the shortcut **chr** of the menu .

**j\_settext**

*procedure j\_settext ( obj: integer; str: PChar ) ;*  
Sets the content or the label of the menu **obj** to **str**.

Menuitem
----------

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

Meter
-------

- j\_meter**                    *function j\_meter ( obj: integer; title: PChar ) : integer;*  
Creates a new pointer-instrument with the specified label **titel**.
- j\_add**                      *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds meter **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to meter **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**                  *procedure j\_disable ( obj: integer ) ;*  
Disables meter **obj** so that it is unresponsive to user interactions
- j\_dispose**                 *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the meter **obj**.
- j\_enable**                   *procedure j\_enable ( obj: integer ) ;*  
enables the meter **obj**.
- j\_focuslistener**         *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to meter **obj**, and returns its event number.
- j\_getdanger**               *procedure j\_getdanger ( obj: integer ) ;*  
Returns the danger value of meter **obj**.
- j\_getfontascent**         *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of meter **obj**.
- j\_getfontheight**         *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of meter **obj**.
- j\_getheight**              *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of meter **obj**.
- j\_getparentid**            *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getparent**               *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getstringwidth**        *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of meter **obj**.
- j\_getwidth**                *function j\_getwidth ( obj: integer ) : integer;*

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



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

Mouselistener
---------------

- j\_mouselistener**      *function j\_mouselistener ( obj: integer; kind: integer ) : integer;*  
Adds a new mouse listener to mouselistener **obj**, and returns its event number.  
An event occurs, if the user action is of kind **kind**.
- j\_dispose**            *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the mouselistener **obj**.
- j\_getmousebutton**    *function j\_getmousebutton ( mouselistener: integer ) : integer;*  
Returns the latest used mousebutton.
- j\_getmousex**         *function j\_getmousex ( mouselistener: integer ) : integer;*  
Returns the current horizontal position of the mouse in its parent's coordinate space.
- j\_getmousey**         *function j\_getmousey ( mouselistener: integer ) : integer;*  
Returns the current vertical position of the mouse in its parent's coordinate space.

Panel
-------

- j\_panel**                    *function j\_panel ( obj: integer ) : integer;*  
Creates a new panel component and returns its event number.
- j\_add**                      *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds panel **obj** to container **cont**
- j\_borderpanel**            *function j\_borderpanel ( obj: integer; type: integer ) : integer;*  
Creates a new borderpanel component with the style **type** and returns its event number.
- j\_button**                   *function j\_button ( obj: integer; label: PChar ) : integer;*  
Creates a new button component with the specified **label** and returns its event number.
- j\_canvas**                   *function j\_canvas ( obj: integer; width: integer; height: integer ) : integer;*  
Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error  $-1$  will be returned.
- j\_checkbox**                *function j\_checkbox ( obj: integer; label: PChar ) : integer;*  
Creates a new checkbox component with the specified **label** and returns its event number.
- j\_choice**                   *function j\_choice ( obj: integer ) : integer;*  
Creates a new choice component and returns its event number.
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to panel **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**                  *procedure j\_disable ( obj: integer ) ;*  
Disables panel **obj** so that it is unresponsive to user interactions
- j\_dispose**                 *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the panel **obj**.
- j\_enable**                   *procedure j\_enable ( obj: integer ) ;*  
enables the panel **obj**.
- j\_focuslistener**           *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to panel **obj**, and returns its event number.
- j\_getfontascent**           *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of panel **obj**.
- j\_getfontheight**           *function j\_getfontheight ( obj: integer ) : integer;*

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

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

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

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

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

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



**j\_showpopup**

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

Printer
---------

- j\_printer**      *function j\_printer ( frame: integer ) : integer;*  
Creates a new object, representing a paper of the printer.
- j\_cliprect**      *procedure j\_cliprect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
Changes current clipping region to the specified rectangle (**x**, **y**, **width**, **height**).
- j\_dispose**      *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the printer **obj**.
- j\_drawarc**      *procedure j\_drawarc ( obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer ) ;*  
Draws an unfilled arc from angle **arc1** to angle **arc2** with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j\_drawcircle**      *procedure j\_drawcircle ( obj: integer; x: integer; y: integer; r: integer ) ;*  
Draws an unfilled circle with center (**x**, **y**) and radius **x**.
- j\_drawimage**      *procedure j\_drawimage ( obj: integer; image: integer; x: integer; y: integer ) ;*  
Copies the image, given by its eventnumber **image**, to position (**x**, **y**).
- j\_drawimagesource**      *procedure j\_drawimagesource ( obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer ) ;*  
Paints an image at Position (**x**, **y**) with **width** and **height**. The red, green and blue values of each pixel are given by the arrays **r**, **g**, **b**.
- j\_drawline**      *procedure j\_drawline ( obj: integer; x1: integer; y1: integer; x2: integer; y2: integer ) ;*  
Draws a line connecting (**x1,y1**) and (**x2,y2**).
- j\_drawoval**      *procedure j\_drawoval ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;*  
Draws an unfilled oval with the center (**x**, **y**) and the horizontal radius **rx** and the vertical radius **ry**.
- j\_drawpixel**      *procedure j\_drawpixel ( obj: integer; x: integer; y: integer ) ;*  
Draws a pixel at (**x,y**).
- j\_drawpolygon**      *procedure j\_drawpolygon ( obj: integer; len: integer; x: var integer; y: var integer ) ;*  
Draws an unfilled polygon based on first **len** elements in **x** and **y**.
- j\_drawpolyline**      *procedure j\_drawpolyline ( obj: integer; len: integer; x: var integer; y: var integer ) ;*  
Draws a series of line segments based on first **len** elements in **x** and **y**.

- j\_drawrect**      *procedure j\_drawrect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
 Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.
- j\_drawroundrect**      *procedure j\_drawroundrect ( obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer ) ;*  
 Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j\_drawscaledimage**      *procedure j\_drawscaledimage ( obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer ) ;*  
 Copy the contents of the rectangular area defined by **x**, **y**,) width **sw**, and height **sh** of the **image** to position **(tx, ty**. The area will be scaled to target width **th** and target height **th**.
- j\_drawstring**      *procedure j\_drawstring ( obj: integer; x: integer; y: integer; str: PChar ) ;*  
 Draws text on screen at position **(x,y)**.
- j\_fillarc**      *procedure j\_fillarc ( obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer ) ;*  
 Draws an filled arc from angle **arc1** to angle **arc2** with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j\_fillcircle**      *procedure j\_fillcircle ( obj: integer; x: integer; y: integer; r: integer ) ;*  
 Draws an filled circle with center **(x, y)** and radius **x**.
- j\_filloval**      *procedure j\_filloval ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;*  
 Draws an filled oval with the center **(x, y)** and the horizontal radius **rx** and the vertical radius **ry**.
- j\_fillpolygon**      *procedure j\_fillpolygon ( obj: integer; len: integer; x: var integer; y: var integer ) ;*  
 Draws an filled polygon based on first **len** elements in **x** and **y**.
- j\_fillrect**      *procedure j\_fillrect ( obj: integer; x: integer; y: integer; width: integer; height: integer ) ;*  
 Draws an filled rectangle from **(x,y)** of size **width** x **height**.
- j\_fillroundrect**      *procedure j\_fillroundrect ( obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer ) ;*  
 Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.
- j\_print**      *procedure j\_print ( obj: integer ) ;*  
 prints the printer .
- j\_setxor**      *procedure j\_setxor ( obj: integer; bool: boolean ) ;*  
 Changes painting mode to XOR mode, if **bool = true** . In this mode, drawing the same object in the same color at the same location twice has no net effect.

**j\_translate**

*procedure j\_translate ( obj: integer; x: integer; y: integer ) ;*  
Moves the origin of drawing operations to (**x**, **y**).

Progressbar
-------------

- j\_progressbar**      *function j\_progressbar ( obj: integer; orient: integer ) : integer;*  
Creates a new progressbar with the specified **orientation**.
- j\_add**              *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds progressbar **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to progressbar **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**          *procedure j\_disable ( obj: integer ) ;*  
Disables progressbar **obj** so that it is unresponsive to user interactions
- j\_dispose**         *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the progressbar **obj**.
- j\_enable**           *procedure j\_enable ( obj: integer ) ;*  
enables the progressbar **obj**.
- j\_focuslistener**   *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to progressbar **obj**, and returns its event number.
- j\_getfontascent**   *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of progressbar **obj**.
- j\_getfontheight**   *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of progressbar **obj**.
- j\_getheight**        *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of progressbar **obj**.
- j\_getparentid**     *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getparent**        *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getstringwidth**   *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of progressbar **obj**.
- j\_getwidth**         *function j\_getwidth ( obj: integer ) : integer;*  
Returns the width of progressbar **obj**.

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

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

Radiobutton
-------------

- j\_radiobutton**      *function j\_radiobutton ( obj: integer; label: PChar ) : integer;*  
Creates a new radiobutton with the specified **label** and returns its event number.
- j\_add**                *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds radiobutton **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to radiobutton **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**            *procedure j\_disable ( obj: integer ) ;*  
Disables radiobutton **obj** so that it is unresponsive to user interactions
- j\_dispose**            *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the radiobutton **obj**.
- j\_enable**             *procedure j\_enable ( obj: integer ) ;*  
enables the radiobutton **obj**.
- j\_focuslistener**     *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to radiobutton **obj**, and returns its event number.
- j\_getfontascent**     *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of radiobutton **obj**.
- j\_getfontheight**     *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of radiobutton **obj**.
- j\_getheight**         *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of radiobutton **obj**.
- j\_getparentid**       *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getparent**         *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame **-1** will be returned.
- j\_getstate**           *function j\_getstate ( obj: integer ) : integer;*  
Returns true , if radiobutton is selected, false otherwise.
- j\_getstringwidth**    *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of radiobutton **obj**.



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

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

Sevensegment
--------------

- j\_sevensegment**      *function j\_sevensegment ( obj: integer; color: integer ) : integer;*  
Creates a new sevensegment display with the specified color **color**.
- j\_add**                      *procedure j\_add ( obj: integer; cont: integer );*  
Adds sevensegment-component **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to sevensegment-component **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**                      *procedure j\_disable ( obj: integer );*  
Disables sevensegment-component **obj** so that it is unresponsive to user interactions
- j\_dispose**                      *procedure j\_dispose ( obj: integer );*  
Releases the resources of the sevensegment-component **obj**.
- j\_enable**                      *procedure j\_enable ( obj: integer );*  
enables the sevensegment-component **obj**.
- j\_focuslistener**              *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to sevensegment-component **obj**, and returns its event number.
- j\_getfontascent**              *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of sevensegment-component **obj**.
- j\_getfontheight**              *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of sevensegment-component **obj**.
- j\_getheight**                      *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of sevensegment-component **obj**.
- j\_getparentid**                      *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getparent**                      *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getstringwidth**              *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of sevensegment-component **obj**.
- j\_getwidth**                      *function j\_getwidth ( obj: integer ) : integer;*

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

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

Scrollpane
------------

- j\_scrollpane**      *function j\_scrollpane ( obj: integer ) : integer;*  
Creates a new scrollpane component and returns its event number.
- j\_add**              *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds scrollpane **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to scrollpane **obj**, and returns its event number.  
An event occurs, if the user action is of kind **kind**.
- j\_disable**          *procedure j\_disable ( obj: integer ) ;*  
Disables scrollpane **obj** so that it is unresponsive to user interactions
- j\_dispose**         *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the scrollpane **obj**.
- j\_enable**           *procedure j\_enable ( obj: integer ) ;*  
enables the scrollpane **obj**.
- j\_focuslistener**   *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to scrollpane **obj**, and returns its event number.
- j\_getfontascent**   *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of scrollpane **obj**.
- j\_getfontheight**   *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of scrollpane **obj**.
- j\_getheight**       *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of scrollpane **obj**.
- j\_getparentid**     *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getparent**       *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getstringwidth**   *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of scrollpane **obj**.
- j\_getviewportheight** *function j\_getviewportheight ( obj: integer ) : integer;*  
Returns the height of the scrollpane 's **obj** port (the area that is shown)

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

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



Textarea
----------

- j\_textarea**      *function j\_textarea ( obj: integer; rows: integer; columns: integer ) : integer;*  
Creates a new textarea component with the specified number of **rows** **columns** and returns its event number.
- j\_add**            *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds textarea **obj** to container **cont**
- j\_appendtext**    *procedure j\_appendtext ( obj: integer; text: PChar ) ;*  
Appends the given **text** to the **obj** current text.
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
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**         *procedure j\_delete ( obj: integer; start: integer; end: integer ) ;*  
Deletes text from starting position **start** to ending position **end**.
- j\_disable**        *procedure j\_disable ( obj: integer ) ;*  
Disables textarea **obj** so that it is unresponsive to user interactions
- j\_dispose**        *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the textarea **obj**.
- j\_enable**         *procedure j\_enable ( obj: integer ) ;*  
enables the textarea **obj**.
- j\_focuslistener**    *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to textarea **obj**, and returns its event number.
- j\_getcolumns**     *procedure j\_getcolumns ( obj: integer ) ;*  
Gets the number of columns in **obj**.
- j\_getcurpos**      *function j\_getcurpos ( obj: integer ) : integer;*  
Returns the position, in characters, of the text cursor.
- j\_getfontascent**    *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of textarea **obj**.
- j\_getfontheight**    *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of textarea **obj**.
- j\_getheight**      *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of textarea **obj**.
- j\_getlength**      *function j\_getlength ( obj: integer ) : integer;*

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

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

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

Textfield
-----------

- j\_textfield**      *function j\_textfield ( obj: integer; columns: integer ) : integer;*  
Creates a new textfield component with the specified number of **columns** and returns its event number.
- j\_add**            *procedure j\_add ( obj: integer; cont: integer );*  
Adds textfield **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to textfield **obj**, and returns its event number. An event occurs, if the user action is of kind **kind**.
- j\_disable**        *procedure j\_disable ( obj: integer );*  
Disables textfield **obj** so that it is unresponsive to user interactions
- j\_dispose**        *procedure j\_dispose ( obj: integer );*  
Releases the resources of the textfield **obj**.
- j\_enable**         *procedure j\_enable ( obj: integer );*  
enables the textfield **obj**.
- j\_focuslistener**    *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to textfield **obj**, and returns its event number.
- j\_getcolumns**      *procedure j\_getcolumns ( obj: integer );*  
Gets the number of columns in **obj**.
- j\_getcurpos**        *function j\_getcurpos ( obj: integer ) : integer;*  
Returns the position, in characters, of the text cursor.
- j\_getfontascent**    *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of textfield **obj**.
- j\_getfontheight**    *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of textfield **obj**.
- j\_getheight**        *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of textfield **obj**.
- j\_getlength**        *function j\_getlength ( obj: integer ) : integer;*  
Returns the length of textfield 's label or text.
- j\_getparentid**      *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

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

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

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



Vscrollbar
------------

- j\_vscrollbar**      *function j\_vscrollbar ( obj: integer ) : integer;*  
Creates a new vertical scrollbar and returns its event number.
- j\_add**              *procedure j\_add ( obj: integer; cont: integer ) ;*  
Adds vscrollbar **obj** to container **cont**
- j\_componentlistener** *function j\_componentlistener ( obj: integer; kind: integer ) : integer;*  
Adds a new componentlistener to vscrollbar **obj**, and returns its event number.  
An event occurs, if the user action is of kind **kind**.
- j\_disable**          *procedure j\_disable ( obj: integer ) ;*  
Disables vscrollbar **obj** so that it is unresponsive to user interactions
- j\_dispose**         *procedure j\_dispose ( obj: integer ) ;*  
Releases the resources of the vscrollbar **obj**.
- j\_enable**           *procedure j\_enable ( obj: integer ) ;*  
enables the vscrollbar **obj**.
- j\_focuslistener**   *function j\_focuslistener ( obj: integer ) : integer;*  
Adds a new focus listener to vscrollbar **obj**, and returns its event number.
- j\_getfontascent**   *function j\_getfontascent ( obj: integer ) : integer;*  
Returns the ascent (space above the baseline) of the actual font of vscrollbar **obj**.
- j\_getfontheight**   *function j\_getfontheight ( obj: integer ) : integer;*  
Returns the total pixel height of the actual font of vscrollbar **obj**.
- j\_getheight**        *function j\_getheight ( obj: integer ) : integer;*  
Returns the height of vscrollbar **obj**.
- j\_getparentid**     *function j\_getparentid ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getparent**        *function j\_getparent ( obj: integer ) : integer;*  
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.
- j\_getstringwidth**   *function j\_getstringwidth ( obj: integer; str: PChar ) : integer;*  
Returns the length of **str** of the actual font of vscrollbar **obj**.
- j\_getvalue**         *function j\_getvalue ( obj: integer ) : integer;*  
Returns the current setting of the scrollbar.

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

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

Window
--------

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

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

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

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

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



# Kapitel 2

## Functions

### additem

Synopsis            procedure **j\_additem** ( obj: integer; str: PChar ) ;

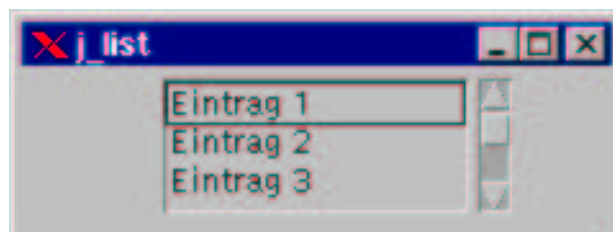
Arguments        obj            integer  
                  str            PChar

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

Targets           List, Choice

Example

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



add
-----

Synopsis	procedure <b>j_add</b> ( obj: integer; cont: integer ) ;
Arguments	obj            integer cont           integer
Description	Adds component <b>obj</b> to container <b>cont</b>
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

## alertbox

Synopsis            procedure **j\_alertbox** ( obj: integer; title: PChar; text: PChar;  
                         button: PChar ) ;

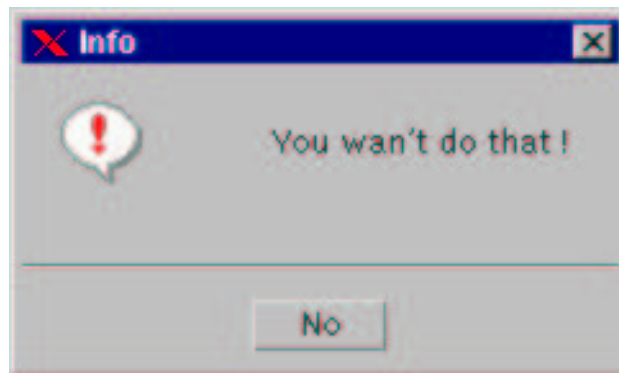
Arguments        obj            integer  
                  title          PChar  
                  text          PChar  
                  button        PChar

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

Targets            Frame

Example

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



appendtext
------------

Synopsis            procedure **j\_appendtext** ( obj: integer; text: PChar ) ;

Arguments        obj            integer  
                  text            PChar

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

Targets            Textarea

beep
------

Synopsis        `procedure j_beep ;`

Description    Emits an audio beep.

## borderpanel

Synopsis            function **j\_borderpanel** ( obj: integer; type: integer ) : integer;

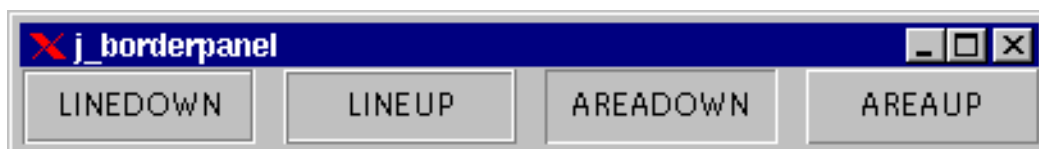
Arguments        obj            integer  
                  type           integer

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

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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



## button

Synopsis	function <b>j_button</b> ( obj: integer; label: PChar ) : integer;
Arguments	obj           integer label         PChar
Description	Creates a new button component with the specified <b>label</b> and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame

### Example

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



canvas
--------

Synopsis      function **j\_canvas** ( obj: integer; width: integer; height: integer  
                 ) : integer;

Arguments    obj            integer  
              width        integer  
              height      integer

Description    Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error  $-1$  will be returned.

Targets        Panel, Borderpanel, Window, Dialog, Frame

Example

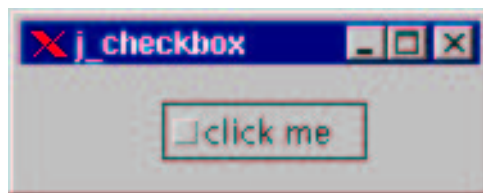
```
:  
canvas := j_canvas(frame,200,50);  
j_setnamedcolorbg(canvas,J_RED);  
:
```





## checkbox

Synopsis	<code>function <b>j_checkbox</b> ( obj: integer; label: PChar ) : integer;</code>
Arguments	<code>obj</code> <code>integer</code> <code>label</code> <code>PChar</code>
Description	Creates a new checkbox component with the specified <b>label</b> and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: frame := j_frame('j_checkbox'); checkbox := j_checkbox(frame, 'click me'); :</pre>



## checkmenuitem

Synopsis      function **j\_checkmenuitem** ( obj: integer; label: PChar ) :  
                 integer;

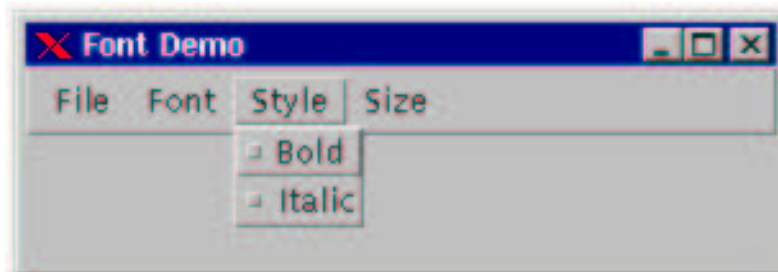
Arguments    obj            integer  
              label          PChar

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\_choicebox2** ( obj: integer; title: PChar; text: PChar; button1: PChar; button2: PChar );

**Arguments**

obj	integer
title	PChar
text	PChar
button1	PChar
button2	PChar

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

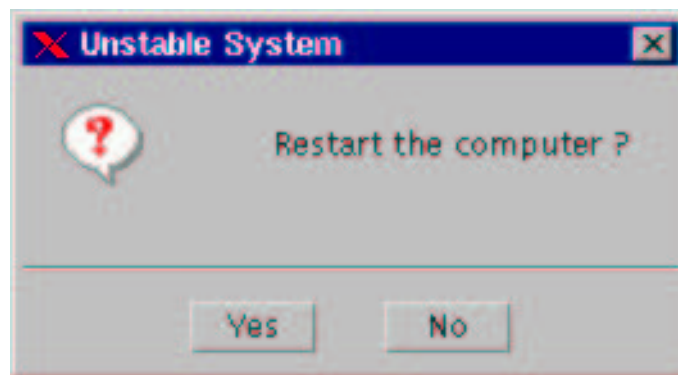
**Targets**            Frame

**Example**

```

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

```



## checkbox3

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

Arguments

obj	integer
title	PChar
text	PChar
button1	PChar
button2	PChar
button3	PChar

Description

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

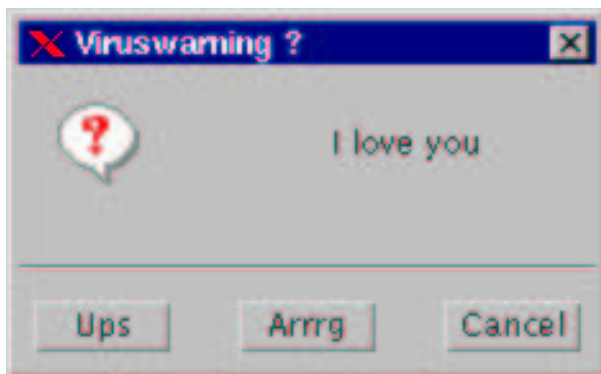
Targets            Frame

Example

```

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

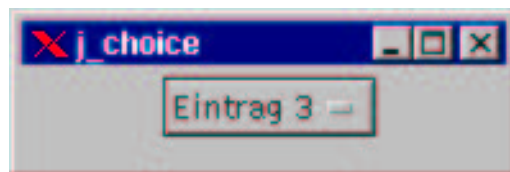
```



## choice

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

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



cliprect
----------

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

componentlistener
-------------------

Synopsis	function <b>j_componentlistener</b> ( obj: integer; kind: integer ) : integer;
Arguments	obj            integer kind           integer
Description	Adds a new componentlistener to component <b>obj</b> , and returns its event number. An event occurs, if the user action is of kind <b>kind</b> . Possible values for <b>kind</b> : <ul style="list-style-type: none"><li>• <b>J_RESIZED</b> : An event occurs when the component has been resized.</li><li>• <b>J_HIDDEN</b> : An event occurs when the component has been hidden.</li><li>• <b>J_SHOWN</b> : An event occurs when the component has been shown.</li></ul>
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensgment

connect
---------

Synopsis           function **j\_connect** ( hostname: PChar ) : boolean;

Arguments       hostname   PChar

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
--------

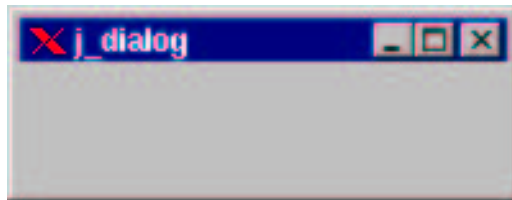
Synopsis	procedure <b>j_delete</b> ( obj: integer; start: integer; end: integer ) ;
Arguments	obj           integer start         integer end           integer
Description	Deletes text from starting position <b>start</b> to ending position <b>end</b> .
Targets	Textarea

deselect
----------

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

## dialog

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



disable
---------

Synopsis	procedure <b>j_disable</b> ( obj: integer ) ;
Arguments	obj            integer
Description	Disables component <b>obj</b> 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 <b>j_dispose</b> ( obj: integer ) ;
Arguments	obj            integer
Description	Releases the resources of the component <b>obj</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphicalabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Canvas, Image, Printer, Keylistener, Focuslistener, Mouselistener

drawarc

Synopsis            procedure **j\_drawarc** ( obj: integer; x: integer; y: integer; rx:  
integer; ry: integer; arc1: integer; arc2: integer ) ;

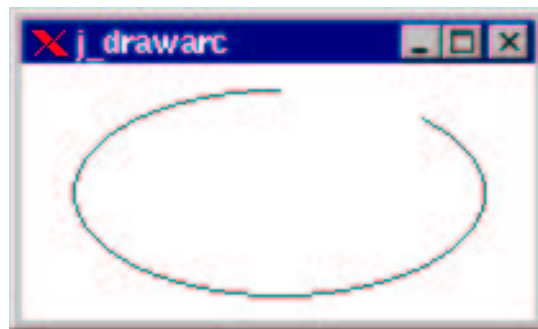
Arguments        obj            integer  
                  x            integer  
                  y            integer  
                  rx            integer  
                  ry            integer  
                  arc1         integer  
                  arc2         integer

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

Targets           Canvas, Image, Printer

Example

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



## drawcircle

Synopsis      procedure **j\_drawcircle** ( obj: integer; x: integer; y: integer; r:  
integer ) ;

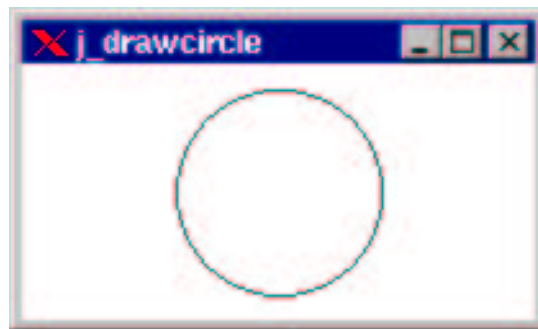
Arguments    obj          integer  
              x          integer  
              y          integer  
              r          integer

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

Targets        Canvas, Image, Printer

Example

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



drawimagesource
-----------------

Synopsis	procedure <b>j_drawimagesource</b> ( obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer ) ;
Arguments	obj            integer x              integer y              integer w              integer h              integer r              var integer (call by reference of the 1st array element) g              var integer (call by reference of the 1st array element) b              var integer (call by reference of the 1st array element)
Description	Paints an image at Position ( <b>x</b> , <b>y</b> ,) with <b>width</b> and <b>height</b> . The red, green and blue values of each pixel are given by the arrays <b>r</b> , <b>g</b> , <b>b</b> .
Targets	Canvas, Image, Printer



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

Synopsis	procedure <b>j_drawimage</b> ( obj: integer; image: integer; x: integer; y: integer ) ;
Arguments	obj           integer image         integer x             integer y             integer
Description	Copies the image, given by its eventnumber <b>image</b> , to position ( <b>x</b> , <b>y</b> ).
Targets	Canvas, Image, Printer

drawline

Synopsis      procedure **j\_drawline** ( obj: integer; x1: integer; y1: integer; x2:  
integer; y2: integer ) ;

Arguments    obj            integer  
              x1            integer  
              y1            integer  
              x2            integer  
              y2            integer

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

Targets        Canvas, Image, Printer

Example

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



## drawoval

Synopsis      procedure **j\_drawoval** ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;

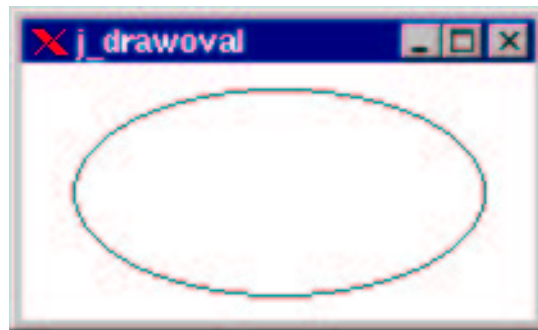
Arguments    obj          integer  
              x          integer  
              y          integer  
              rx         integer  
              ry         integer

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

Targets        Canvas, Image, Printer

Example

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



drawpixel

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

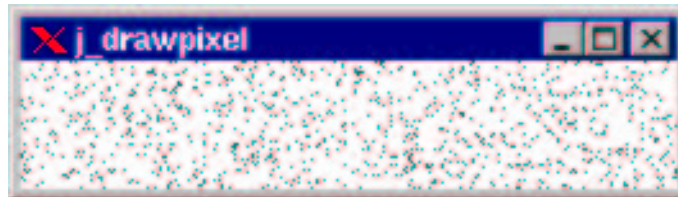
Arguments        `obj            integer`  
                  `x              integer`  
                  `y              integer`

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

Targets           `Canvas, Image, Printer`

Example

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



## drawpolygon

**Synopsis**            procedure **j\_drawpolygon** ( obj: integer; len: integer; x: var integer; y: var integer ) ;

**Arguments**

obj	integer
len	integer
x	var integer (call by reference of the 1st array element)
y	var integer (call by reference of the 1st array element)

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

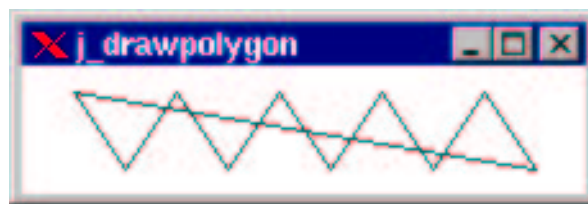
**Targets**             Canvas, Image, Printer

**Example**

```

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

```



## drawpolyline

Synopsis      procedure **j\_drawpolyline** ( obj: integer; len: integer; x: var integer; y: var integer ) ;

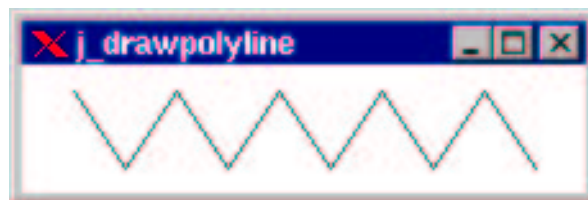
Arguments    obj          integer  
              len          integer  
              x            var integer (call by reference of the 1st array element)  
              y            var integer (call by reference of the 1st array element)

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

Targets        Canvas, Image, Printer

Example

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



## drawrect

Synopsis      procedure **j\_drawrect** ( obj: integer; x: integer; y: integer;  
                 width: integer; height: integer );

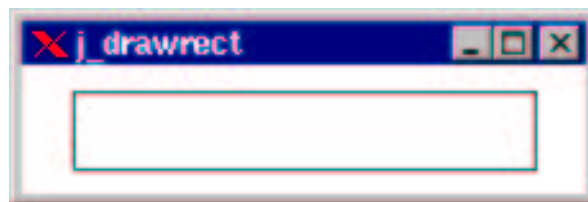
Arguments    obj            integer  
              x            integer  
              y            integer  
              width        integer  
              height       integer

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

Targets        Canvas, Image, Printer

Example

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



## drawroundrect

Synopsis      procedure **j\_drawroundrect** ( obj: integer; x: integer; y: integer; width: integer; height: integer; arcx: integer; arcy: integer )  
;

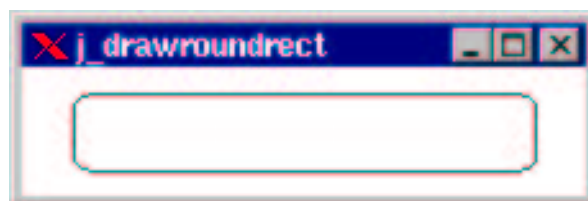
Arguments    obj            integer  
              x            integer  
              y            integer  
              width        integer  
              height       integer  
              arcx         integer  
              arcy         integer

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

Targets        Canvas, Image, Printer

Example

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





drawscaledimage
-----------------

Synopsis	<pre>procedure <b>j_drawscaledimage</b> ( obj: integer; image: integer; sx: integer; sy: integer; sw: integer; sh: integer; tx: integer; ty: integer; tw: integer; th: integer ) ;</pre>																				
Arguments	<table> <tr><td>obj</td><td>integer</td></tr> <tr><td>image</td><td>integer</td></tr> <tr><td>sx</td><td>integer</td></tr> <tr><td>sy</td><td>integer</td></tr> <tr><td>sw</td><td>integer</td></tr> <tr><td>sh</td><td>integer</td></tr> <tr><td>tx</td><td>integer</td></tr> <tr><td>ty</td><td>integer</td></tr> <tr><td>tw</td><td>integer</td></tr> <tr><td>th</td><td>integer</td></tr> </table>	obj	integer	image	integer	sx	integer	sy	integer	sw	integer	sh	integer	tx	integer	ty	integer	tw	integer	th	integer
obj	integer																				
image	integer																				
sx	integer																				
sy	integer																				
sw	integer																				
sh	integer																				
tx	integer																				
ty	integer																				
tw	integer																				
th	integer																				
Description	<p>Copy the contents of the rectangular area defined by <b>x, y,</b> width <b>sw,</b> and height <b>sh</b> of the <b>image</b> to position (<b>tx, ty</b>). The area will be scaled to target width <b>th</b> and target height <b>th</b>.</p>																				
Targets	Canvas, Image, Printer																				

## drawstring

Synopsis            procedure **j\_drawstring** ( obj: integer; x: integer; y: integer;  
                         str: PChar ) ;

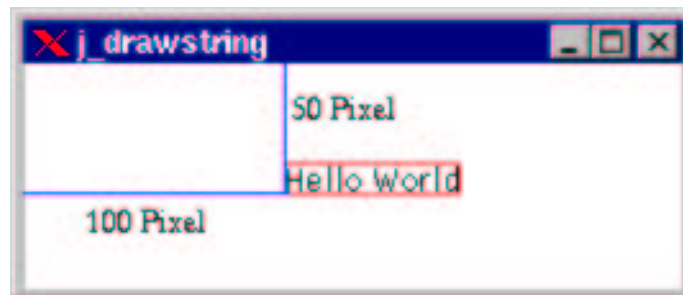
Arguments        obj            integer  
                  x            integer  
                  y            integer  
                  str           PChar

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

Targets           Canvas, Image, Printer

Example

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

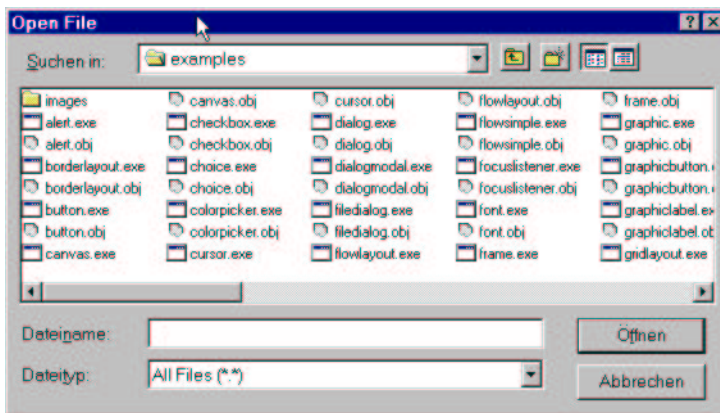


enable
--------

Synopsis	procedure <b>j_enable</b> ( obj: integer ) ;
Arguments	obj            integer
Description	enables the component <b>obj</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensgment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

filedialog

Synopsis	function <b>j_filedialog</b> ( frame: integer; title: PChar; directory: PChar; filename: PChar ) : PChar;								
Arguments	<table border="0"> <tr><td>frame</td><td>integer</td></tr> <tr><td>title</td><td>PChar</td></tr> <tr><td>directory</td><td>PChar</td></tr> <tr><td>filename</td><td>var PChar</td></tr> </table>	frame	integer	title	PChar	directory	PChar	filename	var PChar
frame	integer								
title	PChar								
directory	PChar								
filename	var PChar								
Description	Opens a filedialog box in the specified <b>directory</b> with the specified <b>title</b> and returns the selected <b>filename</b> . If <b>title</b> contains <b>"/S"</b> the SAVE-filedialog will be called. The substring <b>"/S"</b> will be removed.								
Targets	Frame								
Example	<pre> : filename := j_filedialog(frame,'Save/S File','..',filename); : </pre>								



## fileselect

**Synopsis**            function **j\_fileselect** ( frame: integer; title: PChar; filter: PChar;  
filename: PChar ) : PChar;

**Arguments**

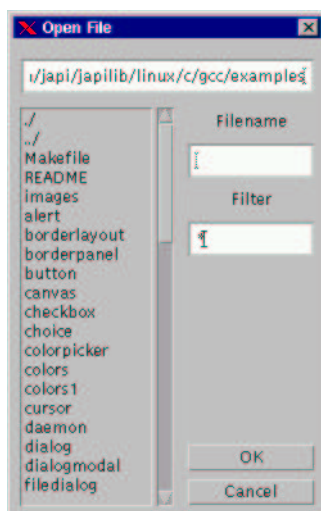
frame	integer
title	PChar
filter	PChar
filename	var PChar

**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 redirections via `j_connect()`;

**Targets**            Frame

**Example**

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



fillarc

Synopsis            procedure **j\_fillarc** ( obj: integer; x: integer; y: integer; rx: integer; ry: integer; arc1: integer; arc2: integer ) ;

Arguments

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

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

Targets            Canvas, Image, Printer

Example

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



## fillcircle

Synopsis      procedure **j\_fillcircle** ( obj: integer; x: integer; y: integer; r:  
integer ) ;

Arguments    obj          integer  
              x          integer  
              y          integer  
              r          integer

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

Targets        Canvas, Image, Printer

Example

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



filloval
----------

Synopsis      procedure **j\_filloval** ( obj: integer; x: integer; y: integer; rx: integer; ry: integer ) ;

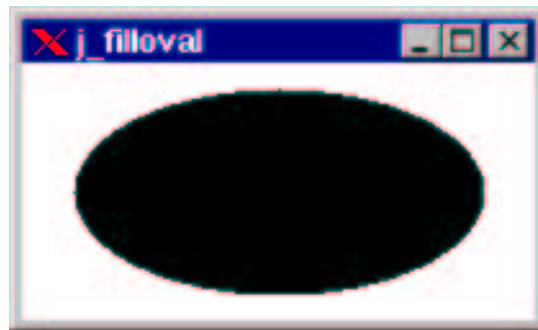
Arguments    obj            integer  
              x            integer  
              y            integer  
              rx           integer  
              ry           integer

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

Targets        Canvas, Image, Printer

Example

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





## fillpolygon

**Synopsis**      procedure **j\_fillpolygon** ( obj: integer; len: integer; x: var integer; y: var integer ) ;

**Arguments**

obj	integer
len	integer
x	var integer (call by reference of the 1st array element)
y	var integer (call by reference of the 1st array element)

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

**Targets**          Canvas, Image, Printer

**Example**

```

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

```



fillrect

Synopsis            procedure **j\_fillrect** ( obj: integer; x: integer; y: integer; width:  
integer; height: integer ) ;

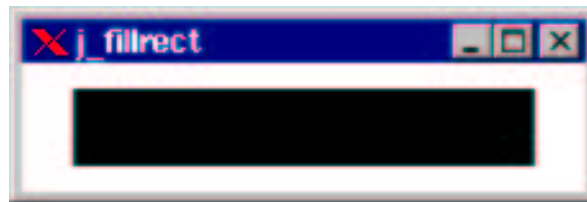
Arguments        obj            integer  
                  x            integer  
                  y            integer  
                  width       integer  
                  height      integer

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

Targets           Canvas, Image, Printer

Example

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



## fillroundrect

**Synopsis**            procedure **j\_fillroundrect** ( obj: integer; x: integer; y: integer;  
width: integer; height: integer; arcx: integer; arcy: integer ) ;

**Arguments**

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

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

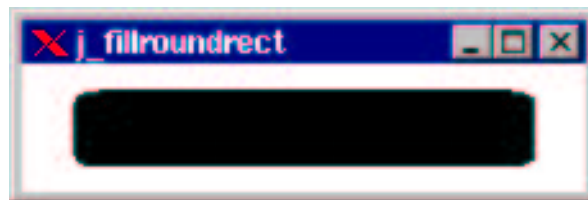
**Targets**             Canvas, Image, Printer

**Example**

```

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

```



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

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

## frame

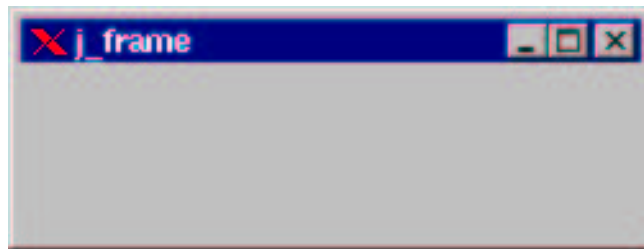
Synopsis            function **j\_frame** ( label: PChar ) : integer;

Arguments        label            PChar

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

Example

```
:  
frame := j_frame('j_frame');  
j_show(frame);  
:
```



getaction
-----------

Synopsis      function **j\_getaction** : integer;

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

## getcolumns

Synopsis        `procedure j_getcolumns ( obj: integer ) ;`

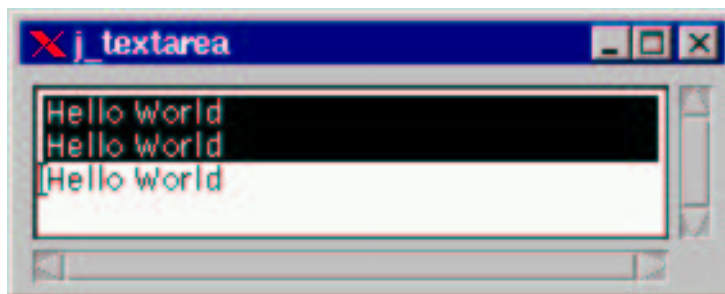
Arguments     `obj`        `integer`

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

Targets        Textarea, Textfield, Gridlayout

Example

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



getcurpos
-----------

Synopsis      function **j\_getcurpos** ( obj: integer ) : integer;

Arguments    obj            integer

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

Targets        Textarea, Textfield



getdanger
-----------

Synopsis            procedure **j\_getdanger** ( obj: integer ) ;

Arguments        obj            integer

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

Targets           Meter

getfontascent
---------------

Synopsis	function <b>j_getfontascent</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Returns the ascent (space above the baseline) of the actual font of component <b>obj</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment

getfontheight
---------------

Synopsis	function <b>j_getfontheight</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Returns the total pixel height of the actual font of component <b>obj</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

getheight
-----------

Synopsis            function **j\_getheight** ( obj: integer ) : integer;

Arguments        obj            integer

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

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

Example

```
:
label := j_getlabel(frame, 'Hello World');
writeln(j_getheight(label));
:
> 22
```

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

Synopsis	function <b>j_getimagesource</b> ( obj: integer; x: integer; y: integer; w: integer; h: integer; r: var integer; g: var integer; b: var integer ) : integer;																
Arguments	<table> <tr><td>obj</td><td>integer</td></tr> <tr><td>x</td><td>integer</td></tr> <tr><td>y</td><td>integer</td></tr> <tr><td>w</td><td>integer</td></tr> <tr><td>h</td><td>integer</td></tr> <tr><td>r</td><td>var integer (call by reference of the 1st array element)</td></tr> <tr><td>g</td><td>var integer (call by reference of the 1st array element)</td></tr> <tr><td>b</td><td>var integer (call by reference of the 1st array element)</td></tr> </table>	obj	integer	x	integer	y	integer	w	integer	h	integer	r	var integer (call by reference of the 1st array element)	g	var integer (call by reference of the 1st array element)	b	var integer (call by reference of the 1st array element)
obj	integer																
x	integer																
y	integer																
w	integer																
h	integer																
r	var integer (call by reference of the 1st array element)																
g	var integer (call by reference of the 1st array element)																
b	var integer (call by reference of the 1st array element)																
Description	Returns an image of the specified size ( <b>x</b> , <b>y</b> , <b>width</b> , <b>height</b> ) of component . The red, green and blue values of each pixel will be stored in <b>r</b> , <b>g</b> , <b>b</b>																
Targets	Canvas, Image																

getimage
----------

Synopsis	function <b>j_getimage</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Copy the contents of component <b>obj</b> into an image and return its eventnumber.
Targets	Canvas, Image

getinheight
-------------

Synopsis           function **j\_getinheight** ( cont: integer ) : integer;

Arguments         cont           integer

Description       Returns the height of the client size.

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
frame := j_frame('Hello World');
j_setsize(frame,300,400);
writeln(j_getinheight(frame));
:
> 370
```

## getinsets

Synopsis	function <b>j_getinsets</b> ( obj: integer; side: integer ) : integer;
Arguments	obj            integer side           integer
Description	Returns the width of the specified inset. <b>side</b> can take the following values: <ul style="list-style-type: none"> <li>• J.TOP: returns the height of the top inset.</li> <li>• J.BOTTOM: returns the height of the bottom inset.</li> <li>• J.LEFT: returns the width of the left inset.</li> <li>• J.RIGHT: returns the width of the right inset.</li> </ul>
Targets	Panel, Borderpanel, Window, Dialog, Frame

### Example

```

:
frame = j_frame('j_getinsets');
writeln(j_getinsets(frame,J_TOP),j_getinsets(frame,J_BOTTOM),
        j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT));
:
> 25 5 5 6

```





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

Synopsis           function **j\_getinwidth** ( cont: integer ) : integer;

Arguments         cont           integer

Description       Returns the width of the client size.

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
frame := j_frame('Hello World');
j_setsize(frame,300,400);
writeln(j_getinwidth(frame));
:
> 289
```

getitemcount
--------------

Synopsis      function **j\_getitemcount** ( obj: integer ) : integer;

Arguments    obj            integer

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

Targets        List, Choice

getitem
---------

Synopsis	function <b>j_getitem</b> ( obj: integer; item: integer; str: PChar ) : PChar;
Arguments	obj           integer item          integer str           var PChar
Description	returns the label of the given <b>item</b> .
Targets	List, Choice

getkeychar
------------

Synopsis      function **j\_getkeychar** ( obj: integer ) : integer;

Arguments    obj            integer

Description   Returns the ascii value of the last pressed key.

Targets        Keylistener

getkeycode
------------

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

getlayoutid
-------------

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

getlength
-----------

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

getmousebutton

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



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

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

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

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

## getparentid

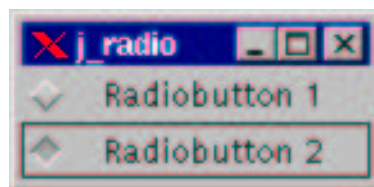
Synopsis	function <b>j_getparentid</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Returns the parent event number of component <b>obj</b> . If <b>obj</b> is a frame <b>-1</b> will be returned.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel,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

### Example

```

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

```



## getparent

Synopsis	function <b>j_getparent</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Returns the parent event number of component <b>obj</b> . If <b>obj</b> is a frame <b>-1</b> will be returned.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Menubar, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Radiogroup
Example	<pre>: radio1        := j_radiobutton(j_radiogroup(frame), 'Radiobutton 1'); radio2        := j_radiobutton(j_getparent(radio1), 'Radiobutton 2'); :</pre>



## getrows

Synopsis        procedure **j\_getrows** ( obj: integer ) ;

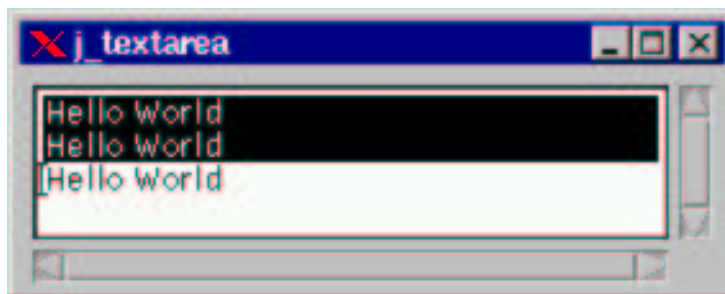
Arguments     obj            integer

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

Targets        Textarea, Gridlayout

Example

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



getscaledimage
----------------

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

getscreenheight
-----------------

Synopsis           function **j\_getscreenheight** : integer;

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

Example

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

getscreenwidth
----------------

Synopsis           function **j\_getscreenwidth** : integer;

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

Example

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



getselect
-----------

Synopsis           function **j\_getselect** ( obj: integer ) : integer;

Arguments       obj           integer

Description      Returns the position of currently selected item.

Targets          List, Choice

getselend
-----------

Synopsis      function **j\_getselend** ( obj: integer ) : integer;

Arguments    obj            integer

Description   Returns the ending position of any selected text.

Targets        Textarea, Textfield

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

Synopsis           function **j\_getselstart** ( obj: integer ) : integer;

Arguments       obj           integer

Description      Returns the initial position of any selected text.

Targets          Textarea, Textfield

getseltext
------------

Synopsis      function **j\_getseltext** ( obj: integer; text: PChar ) : PChar;

Arguments    obj            integer  
              text            var PChar

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

Targets        Textarea, Textfield

getstate
----------

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

getstringwidth

Synopsis	function <b>j_getstringwidth</b> ( obj: integer; str: PChar ) : integer;
Arguments	obj           integer str           PChar
Description	Returns the length of <b>str</b> of the actual font of component <b>obj</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

gettext
---------

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

getvalue
----------

Synopsis           function **j\_getvalue** ( obj: integer ) : integer;

Arguments       obj           integer

Description      Returns the current setting of the scrollbar.

Targets          Scrollbar



getviewportheight
-------------------

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

getviewportwidth
------------------

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

## getwidth

Synopsis	function <b>j_getwidth</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Returns the width of component <b>obj</b> .
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'); writeln(j_getwidth(label)); : &gt; 84</pre>

getxpos
---------

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

getypos
---------

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

## graphicbutton

Synopsis           function **j\_graphicbutton** ( obj: integer; filename: PChar ) :  
integer;

Arguments       obj           integer  
                 filename     PChar

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

Targets          Panel, Borderpanel, Window, Dialog, Frame

Example

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



## graphiclabel

Synopsis            `function j_graphiclabel ( obj: integer; str: PChar ) : integer;`

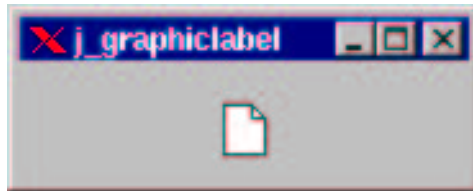
Arguments        `obj            integer`  
                  `str            PChar`

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

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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



hasfocus
----------

Synopsis      `function j_hasfocus ( obj: integer ) : integer;`

Arguments    `obj            integer`

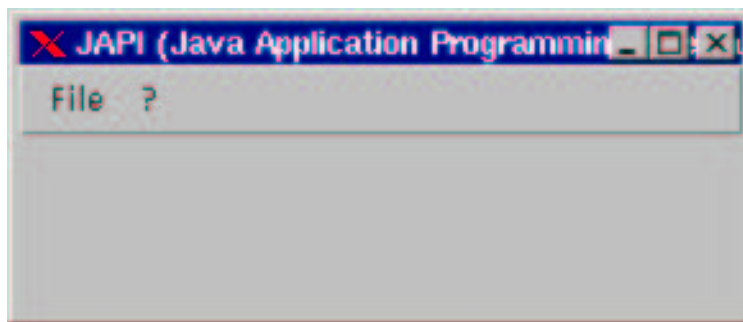
Description    Returns true if the component has the focus, false otherwise.

Targets        Focuslistener



## helpmenu

Synopsis	function <b>j_helpmenu</b> ( obj: integer; label: PChar ) : integer;
Arguments	obj            integer label          PChar
Description	Creates a new helpmenu component with the specified <b>label</b> and returns its event number.
Targets	Menubar
Example	<pre>: frame := j_frame('Menu Komponenten'); menubar := j_menubar(frame); file:= j_menu(menubar,'File'); help:= j_helpmenu(menubar,'?') :</pre>



hide
------

Synopsis      procedure **j\_hide** ( obj: integer ) ;

Arguments    obj            integer

Description   Hides the component **obj**.

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

## hscrollbar

Synopsis	function <b>j_hscrollbar</b> ( obj: integer ) : integer;
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); j_setpos(scroll,20,40); j_setsize(scroll,150,20); :</pre>

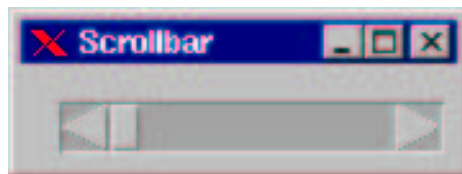


image
-------

Synopsis            function **j\_image** ( width: integer; height: integer ) : integer;

Arguments        width        integer  
                  height       integer

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

Example

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

insert
--------

Synopsis	function <b>j_insert</b> ( obj: integer; pos: integer; label: PChar ) : integer;
Arguments	obj           integer pos           integer label         PChar
Description	inserts a new item to component <b>obj</b> at position <b>pos</b> with the specified <b>label</b> .
Targets	List, Choice

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

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

isparent
----------

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

isresizable
-------------

Synopsis	function <b>j_isresizable</b> ( obj: integer ) : boolean;
Arguments	obj            integer
Description	returns true if component is resizable, false otherwise
Targets	Dialog, Frame



isselect
----------

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

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

Synopsis            `function j_isvisible ( obj: integer ) : integer;`

Arguments        `obj            integer`

Description      Returns true if **obj** is visible, false otherwise.

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

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

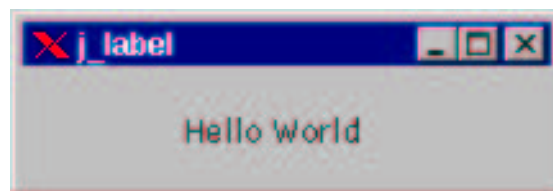
Synopsis	function <b>j_keylistener</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Adds a new key listener to component <b>obj</b> , and returns its event number.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel,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

## label

Synopsis	function <b>j_label</b> ( obj: integer; label: PChar ) : integer;
Arguments	obj           integer label         PChar
Description	Creates a new label component with the specified <b>label</b> and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame

### Example

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



## led

**Synopsis**            `function j_led ( obj: integer; style: integer; color: integer ) : integer;`

**Arguments**

<code>obj</code>	<code>integer</code>
<code>style</code>	<code>integer</code>
<code>color</code>	<code>integer</code>

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

**Targets**            Panel, Borderpanel, Window, Dialog, Frame

**Example**

```

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

```



line

Synopsis           function **j\_line** ( obj: integer; orient: integer; style: integer;  
length: integer ) : integer;

Arguments       obj           integer  
                 orient       integer  
                 style        integer  
                 length      integer

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

- J\_HORIZONTAL : horizontal line
- J\_VERTICAL : vertical line

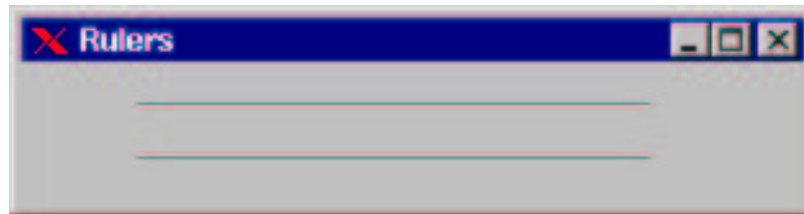
The Parameter **style** specifies the linestyle:

- J\_LINEDOWN : etched-in linestyle.
- J\_LINEUP : etchet-out linestyle.

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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

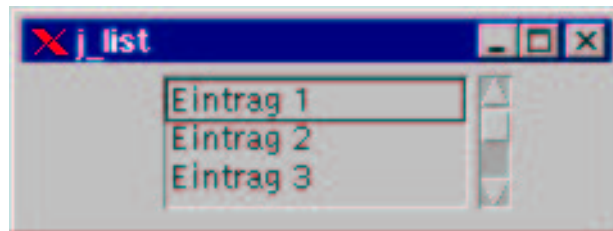


## list

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

### Example

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



loadimage
-----------

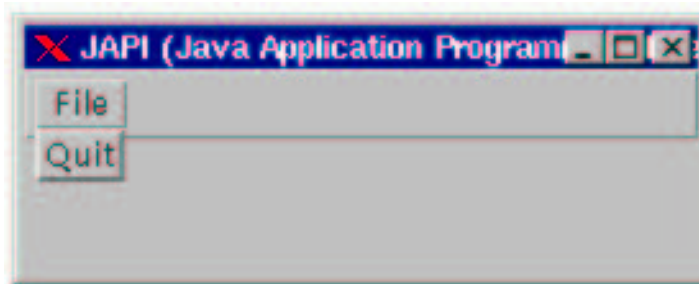
Synopsis	function <b>j_loadimage</b> ( filename: PChar ) : integer;
Arguments	filename    PChar
Description	Loads the Image from file <b>filename</b> and returns its eventnumber. The file could be of the following format: <ul style="list-style-type: none"><li>• GIF</li><li>• JPEG</li><li>• BMP</li><li>•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



## menubar

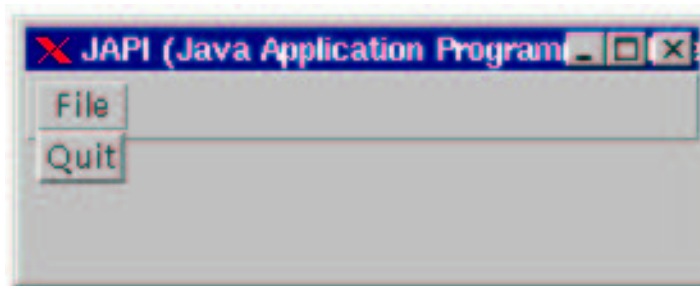
Synopsis	function <b>j_menubar</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Creates a new menubar and returns its event number.
Targets	Frame
Example	

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



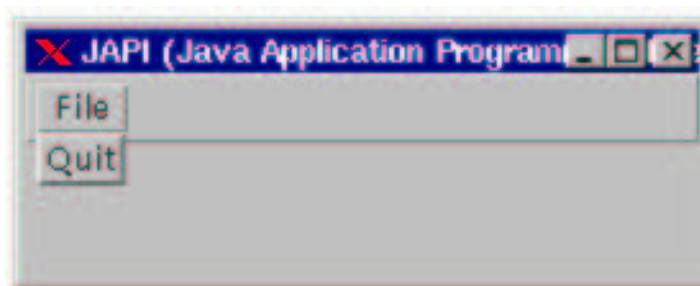
## menuItem

Synopsis	function <b>j_menuitem</b> ( obj: integer; label: PChar ) : integer;
Arguments	obj           integer label         PChar
Description	Creates a new menuItem with the specified <b>label</b> and returns its event number.
Targets	Menu, Popupmenu, Helpmenu
Example	<pre>: frame := j_frame('Menu Komponenten'); menubar := j_menubar(frame); file := j_menu(menubar, 'File'); quit := j_menuitem(file, 'Quit'); :</pre>



menu
------

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



messagebox

Synopsis            procedure **j\_messagebox** ( obj: integer; title: PChar; text:  
                         PChar ) ;

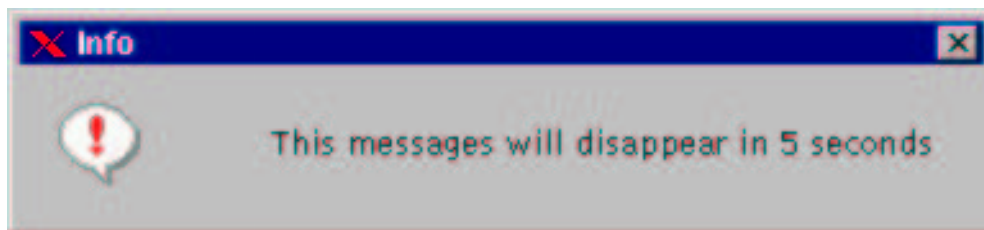
Arguments        obj            integer  
                         title          PChar  
                         text          PChar

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

Targets            Frame

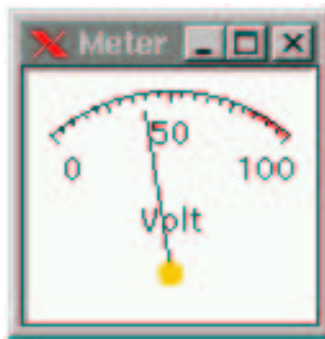
Example

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



## meter

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



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

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

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

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

nextaction
------------

Synopsis      function **j\_nextaction** : integer;

Description      Waits for the next event.



## pack

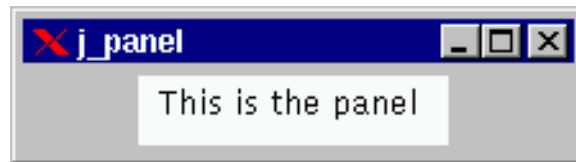
Synopsis	procedure <b>j_pack</b> ( obj: integer ) ;
Arguments	obj            integer
Description	Resizes component to the minimal size of contained components.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	<pre>: j_setflowlayout(jframe,J_HORIZONTAL); canvas := j_canvas(frame,200,50); j_setnamedcolorbg(canvas,J_RED);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



## panel

Synopsis	function <b>j_panel</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Creates a new panel component and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	

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



## popupmenu

Synopsis            `function j_popupmenu ( obj: integer; label: PChar ) : integer;`

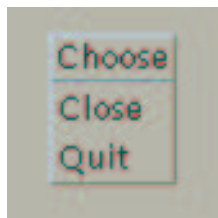
Arguments        `obj            integer`  
                  `label         PChar`

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

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

Example

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



printer
---------

Synopsis           function **j\_printer** ( frame: integer ) : integer;

Arguments        frame        integer

Description      Creates a new object, representing a paper of the printer and returns its event number. On error  $-1$  will be returned. A printer object can be used like a canvas, where all drawing funktions will be passed to the printer, instead of a window. A printer generates no event.

Targets          Frame

Example

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

## print

Synopsis	procedure <b>j_print</b> ( obj: integer ) ;
Arguments	obj            integer
Description	prints the component . With X-Windows all components have Motif-look. If component is a printer, the actual page will be closed, and a new page will be opened. The pages are not jet printed. To print all pages call j_dispose(printer);
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, Canvas, Image, Printer

### Example

```

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

```



## progressbar

Synopsis            function **j\_progressbar** ( obj: integer; orient: integer ) : integer;

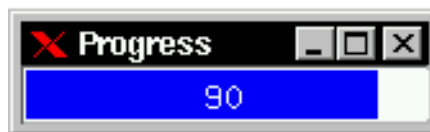
Arguments        obj            integer  
                  orient        integer

Description      Creates a new progressbar with the specified **orientation** and returns its event number. Orientation could be J\_HORIZONTAL or J\_VERTICAL. The progressbar has predefined values from 0 to 100. This can be changed via *j\_setmin()* and *j\_setmax()*.

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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



quit
------

Synopsis        `procedure j_quit ;`

Description    Cancels the connection to the JAPI Kernel.

## radiobutton

Synopsis            `function j_radiobutton ( obj: integer; label: PChar ) : integer;`

Arguments        `obj            integer`  
                  `label          PChar`

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

Targets           Radiogroup

Example

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





## radiogroup

Synopsis            `function j_radiogroup ( obj: integer ) : integer;`

Arguments        `obj            integer`

Description       `Creates a new radiogroup and returns its event number.`

Targets           `Panel, Borderpanel, Window, Dialog, Frame`

Example

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



random
--------

Synopsis

function **j\_random** : integer;

Description

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

releaseall
------------

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

release
---------

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

removeall
-----------

Synopsis      function **j\_removeall** ( obj: integer ) : integer;

Arguments    obj            integer

Description   Removes all items from the component .

Targets        List, Choice

removeitem
------------

Synopsis      function **j\_removeitem** ( obj: integer; item: PChar ) : integer;

Arguments    obj            integer  
              item            PChar

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

Targets        List, Choice

remove
--------

Synopsis           function **j\_remove** ( obj: integer; item: integer ) : integer;

Arguments       obj           integer  
                 item         integer

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

Targets          List, Choice

replacetext
-------------

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



saveimage
-----------

Synopsis           function **j\_saveimage** ( obj: integer; filename: PChar; filetype:  
integer ) : boolean;

Arguments         obj           integer  
                  filename     PChar  
                  filetyp     integer

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

- J\_BMP Win32 Bitmap Format
- J\_PPM Portable pixmap

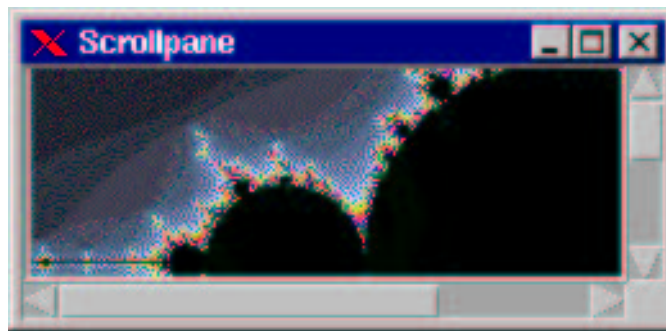
Example

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

## scrollpane

Synopsis	function <b>j_scrollpane</b> ( obj: integer ) : integer;
Arguments	obj            integer
Description	Creates a new scrollpane component and returns its event number.
Targets	Panel, Borderpanel, Window, Dialog, Frame
Example	

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



selectall
-----------

Synopsis            procedure **j\_selectall** ( obj: integer ) ;

Arguments        obj            integer

Description      Selects all the text in the component .

Targets           Textarea, Textfield

select
--------

Synopsis           function **j\_select** ( obj: integer; item: integer ) : integer;

Arguments       obj           integer  
                  item          integer

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

Targets          List, Choice

selecttext
------------

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

## seperator

Synopsis            `procedure j_seperator ( obj: integer ) ;`

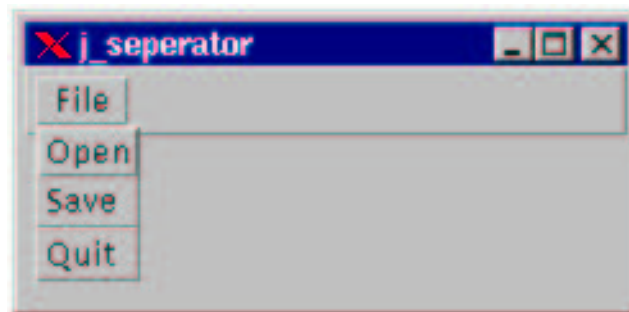
Arguments        `obj            integer`

Description       `Adds a separator bar to the component .`

Targets           `Menu, HelpMenu, Popupmenu`

Example

```
:  
file := j_menu(menubar, 'File');  
open := j_menuitem(file, 'Open');  
save := j_menuitem(file, 'Save');  
j_seperator(file);  
quit := j_menuitem(file, 'Quit');  
:
```



setalign
----------

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

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

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



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

Synopsis            procedure **j\_setborderlayout** ( obj: integer ) ;

Arguments        obj            integer

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

Targets           Panel, Borderpanel, Window, Dialog, Frame

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

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

## setcolorbg

**Synopsis**            procedure **j\_setcolorbg** ( obj: integer; r: integer; g,: integer; b:  
integer ) ;

**Arguments**        obj            integer  
                      r                integer  
                      g,             integer  
                      b                integer

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

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

**Example**

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



setcolor
----------

Synopsis            procedure **j\_setcolor** ( obj: integer; r: integer; g,: integer; b:  
integer ) ;


Arguments        obj            integer  
                  r                integer  
                  g,             integer  
                  b                integer

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

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

Example

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



## setcolumns

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

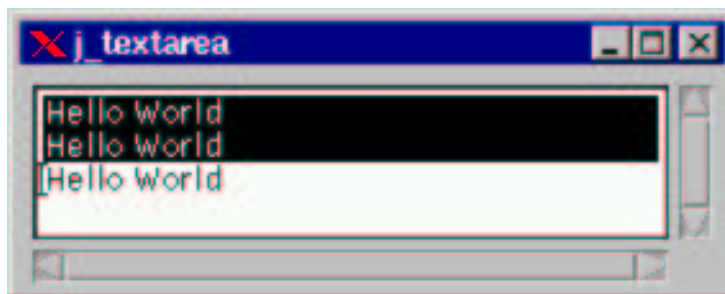
Arguments        `obj            integer`  
                  `columns      integer`

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

Targets           Textarea, Textfield, Gridlayout

Example

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



setcurpos
-----------

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

setcursor
-----------

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

setdanger
-----------

Synopsis      procedure **j\_setdanger** ( obj: integer; val: integer ) ;

Arguments    obj          integer  
              val          integer

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

Targets        Meter



setdebug
----------

Synopsis            procedure **j\_setdebug** ( level: integer ) ;

Arguments        level            integer

Description      Sets the debuglevel to **level**.

setechochar
-------------

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

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

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

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

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

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

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

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

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

setfocus
----------

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

setfontname
-------------

Synopsis            procedure **j\_setfontname** ( obj: integer; name: integer ) ;

Arguments        obj            integer  
                  name           integer

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

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

Example

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



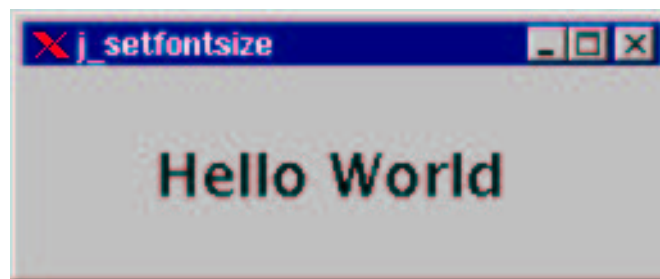


## setfontsize

Synopsis	procedure <b>j_setfontsize</b> ( obj: integer; size: integer ) ;
Arguments	obj            integer size           integer
Description	Changes the font to the given <b>size</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu

### Example

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



setfontstyle

Synopsis	procedure <b>j_setfontstyle</b> ( obj: integer; style: integer ) ;
Arguments	obj            integer style          integer
Description	Changes the font to the given <b>style</b> .
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu
Example	: label := j_label(jframe,'Hello World'); j_setfontstyle(label,J_BOLD+J_ITALIC); :



## setfont

Synopsis	procedure <b>j_setfont</b> ( obj: integer; name: integer; style: integer; size: integer ) ;								
Arguments	<table border="0" style="margin-left: 20px;"> <tr><td>obj</td><td>integer</td></tr> <tr><td>name</td><td>integer</td></tr> <tr><td>style</td><td>integer</td></tr> <tr><td>size</td><td>integer</td></tr> </table>	obj	integer	name	integer	style	integer	size	integer
obj	integer								
name	integer								
style	integer								
size	integer								
Description	Changes the font to the given characteristics <b>name</b> , <b>style</b> and <b>size</b> .								
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevenssegment, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu								
Example	<pre> : label := j_label(jframe,'Hello World'); j_setfont(label,J_TIMES,J_PLAIN,18); : </pre>								

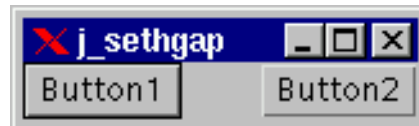


setgridlayout
---------------

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

## sethgap

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



seticon
---------

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

## setimage

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

Arguments        `obj            integer`  
                  `image        integer`

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

Targets            Graphicbutton, Graphiclabel

Example

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



setinsets

Synopsis            procedure **j\_setinsets** ( obj: integer; top: integer; bottom: integer; left: integer; right: integer ) ;

Arguments        obj            integer  
                   top            integer  
                   bottom       integer  
                   left           integer  
                   right          integer

Description       Set the insets to the specified values.

Targets            Panel, Borderpanel, Window, Dialog, Frame

Example

```

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

```





setmax
--------

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

setmin
--------

Synopsis           function **j\_setmin** ( obj: integer; val: integer ) : integer;

Arguments       obj           integer  
                  val           integer

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

Targets           Scrollbar, Meter, Progress

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

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

setnamedcolor
---------------

Synopsis            procedure **j\_setnamedcolor** ( obj: integer; color: integer ) ;

Arguments        obj            integer  
                  color          integer

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

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

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

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

start
-------

Synopsis            procedure **j\_start** ( port: integer ) ;

Arguments        port            integer

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

Example

```
:
j_setport(12345);
if( not j_start ) then begin
:
```

setpos
--------

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

setradiogroup
---------------

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



setresizable
--------------

Synopsis            procedure **j\_setresizable** ( obj: integer; resizable: boolean ) ;

Arguments        obj            integer  
                  resizable    boolean

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

Targets          Dialog, Frame

Example

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

setrows

Synopsis            procedure **j\_setrows** ( obj: integer; rows: integer ) ;

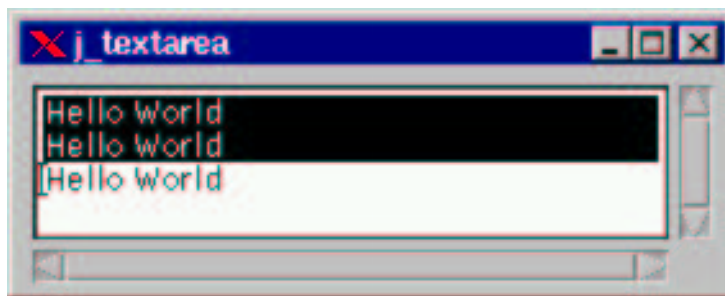
Arguments        obj            integer  
                  rows           integer

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

Targets           Textarea, Gridlayout

Example

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



setshortcut
-------------

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

setsize
---------

Synopsis      procedure **j\_setsize** ( obj: integer; width: integer; height: integer ) ;

Arguments    obj            integer  
              width        integer  
              height      integer

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

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

Example

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



setslidesize
--------------

Synopsis           function **j\_setslidesize** ( obj: integer; val: integer ) : integer;

Arguments       obj           integer  
                  val           integer

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

Targets          Scrollbar

setstate
----------

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


Arguments        `obj            integer`  
                  `bool            boolean`

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

Targets            Checkbox, Radiobutton, CheckmenuItem, Led

## settext

Synopsis	procedure <b>j_settext</b> ( obj: integer; str: PChar ) ;
Arguments	obj            integer str            PChar
Description	Sets the content or the label of the component <b>obj</b> to <b>str</b> .
Targets	Button, Label, Checkbox, Radiobutton, Dialog, Frame, MenuItem, CheckBoxMenuItem, Menu, HelpMenu, Popupmenu, Textarea, Textfield
Example	<pre>: button := j_button(frame,'Hello World'); j_settext(button,'Goodbye'); :</pre>



Goodbye

setunitinc
------------

Synopsis      function **j\_setunitinc** ( obj: integer; val: integer ) : integer;

Arguments    obj          integer  
              val          integer

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

Targets        Scrollbar



setvalue
----------

Synopsis            procedure **j\_setvalue** ( obj: integer; val: integer ) ;

Arguments        obj            integer  
                  val            integer

Description      Changes the current value of the component to **val**.

Targets           Scrollbar, Progress, Meter, Sevensegment

## setvgap

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



setxor
--------

Synopsis	procedure <b>j_setxor</b> ( obj: integer; bool: boolean ) ;
Arguments	obj            integer bool           boolean
Description	Changes painting mode to XOR mode, if bool = true . In this mode, drawing the same object in the same color at the same location twice has no net effect.
Targets	Canvas, Image, Printer

## sevenssegment

Synopsis      function **j\_sevenssegment** ( obj: integer; color: integer ) : integer;

Arguments    obj            integer  
              color        integer

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

Targets        Panel, Borderpanel, Window, Dialog, Frame

Example

```
:  
seven := j_sevenssegment(frame, J_GREEN);  
j_setvalue(seven, 5);  
:
```



showpopup
-----------

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

show
------

Synopsis        procedure **j\_show** ( obj: integer ) ;

Arguments     obj            integer

Description    Shows the component **obj**.

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

sleep
-------

Synopsis           function **j\_sleep** ( msec: integer ) : integer;

Arguments        msec           integer

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

start

Synopsis           function **j\_start** : boolean;

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

Example

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



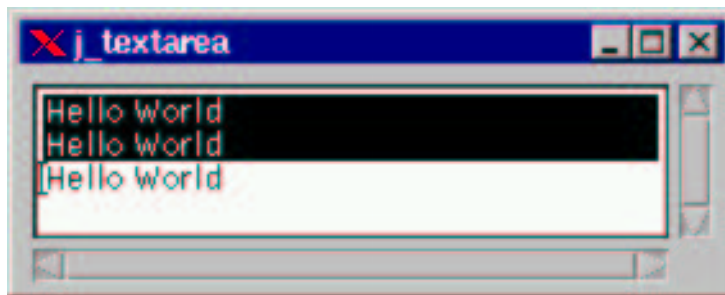
sync
------

Synopsis        `procedure j_sync ;`

Description    Synchronizes the application with the JAPI kernel.

textarea

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



## textfield

Synopsis            `function j_textfield ( obj: integer; columns: integer ) : integer;`

Arguments        `obj            integer`  
                  `columns       integer`

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

Targets           Panel, Borderpanel, Window, Dialog, Frame

Example

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



translate
-----------

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

## vscrollbar

Synopsis	function <b>j_vscrollbar</b> ( obj: integer ) : integer;
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); j_setpos(scroll,120,40); j_setsize(scroll,20,100); :</pre>



windowlistener
----------------

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

## window

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

Mouse pressed at 108:179