japi

Reference Manual

Merten Joost

Inhaltsverzeichnis

Teil I

Reference

Kapitel 1

Components

Button

j_button *integer function j_button (integer obj , character*(*) label)*
Creates a new button component with the specified **label** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds button **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to button **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables button **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the button **obj**.

j_enable *procedure j_enable (integer obj)*
enables the button **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to button **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of button **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of button **obj**.

j_getheight *integer function j_getheight (integer obj)*

Returns the height of button **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of button 's label or text.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the button 's text or label.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of button **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of button **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of button **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the button **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to button **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to button **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character∗(∗) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the button .

j_release *procedure j_release (integer obj)*
Releases button **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves button **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the button 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to button **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

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

Borderpanel

j_borderpanel *integer function j_borderpanel (integer obj , integer type)*
Creates a new borderpanel component with the style **type** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds borderpanel **obj** to container **cont**

j_borderpanel *integer function j_borderpanel (integer obj , integer type)*
Creates a new borderpanel component with the style **type** and returns its event number.

j_button *integer function j_button (integer obj , character*(*) label)*
Creates a new button component with the specified **label** and returns its event number.

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

j_checkbox *integer function j_checkbox (integer obj , character*(*) label)*
Creates a new checkbox component with the specified **label** and returns its event number.

j_choice *integer function j_choice (integer obj)*
Creates a new choice component and returns its event number.

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to borderpanel **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables borderpanel **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the borderpanel **obj**.

j_enable *procedure j_enable (integer obj)*
enables the borderpanel **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to borderpanel **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of borderpanel **obj**.

j_getfontheight
_integer function j_getfontheight (integer obj)_
Returns the total pixel height of the actual font of borderpanel **obj**.

j_getheight
_integer function j_getheight (integer obj)_
Returns the height of borderpanel **obj**.

j_getinsets
_integer function j_getinsets (integer obj , integer side)_
Returns the width of the specified inset.

j_getlayoutid
_integer function j_getlayoutid (integer obj)_
Returns the event number of the layoutmanager for containers **obj**.

j_getparentid
_integer function j_getparentid (integer obj)_
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent
_integer function j_getparent (integer obj)_
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getwidth
_integer function j_getwidth (integer obj)_
Returns the width of borderpanel **obj**.

j_getxpos
_integer function j_getxpos (integer obj)_
Returns the current horizontal position of borderpanel **obj** in its parent's coordinate space.

j_getypos
_integer function j_getypos (integer obj)_
Returns the current vertical position of borderpanel **obj** in its parent's coordinate space.

j_graphicbutton
_integer function j_graphicbutton (integer obj , character*(*) filename)_
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.

j_graphiclabel
_integer function j_graphiclabel (integer obj , character*(*) str)_
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

j_hide
_procedure j_hide (integer obj)_
Hides the borderpanel **obj**.

j_hscrollbar
_integer function j_hscrollbar (integer obj)_
Creates a new horizontal scrollbar and returns its event number.

j_isparent
_integer function j_isparent (integer obj , integer cont)_
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible
_integer function j_isvisible (integer obj)_
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
_integer function j_keylistener (integer obj)_
Adds a new key listener to borderpanel **obj**, and returns its event number.

j_label *integer function j_label (integer obj , character*(*) label)*
Creates a new label component with the specified **label** and returns its event number.

j_led *integer function j_led (integer obj , integer style , integer color)*
Creates a new led component with the specified **style** and the specified color **color**.

j_line *integer function j_line (integer obj , integer orient , integer style , integer length)*
Creates a new line component with the specified **length** and returns its event number.

j_list *integer function j_list (integer obj , integer rows)*
Creates a new list component with the specified number of **rows** and returns its event number.

j_meter *integer function j_meter (integer obj , character*(*) title)*
Creates a new pointer–intrument with the specified label **titel**.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to borderpanel **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_pack *procedure j_pack (integer obj)*
Resizes borderpanel to the minimal size of contained components.

j_panel *integer function j_panel (integer obj)*
Creates a new panel component and returns its event number.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the borderpanel .

j_progressbar *integer function j_progressbar (integer obj , integer orient)*
Creates a new progressbar with the specified **orient**ation.

j_radiogroup *integer function j_radiogroup (integer obj)*
Creates a new radiogroup and returns its event number.

j_releaseall *procedure j_releaseall (integer obj)*
Releases all components from borderpanel **obj**.

j_release *procedure j_release (integer obj)*
Releases borderpanel **obj** from its parent component (container).

j_scrollpane *integer function j_scrollpane (integer obj)*
Creates a new scrollpane component and returns its event number.

j_setalign *procedure j_setalign (integer obj , integer align)*

Sets the alignment in borderpanel **obj** to **align**. Needs a flowlayout Manager.

j_setborderlayout *procedure j_setborderlayout (integer obj)*
Adds a borderlayout manager to borderpanel **obj**.

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves borderpanel **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the borderpanel 's **obj** cursor to the specified **cursor**.

j_setfixlayout *procedure j_setfixlayout (integer obj)*
Adds a fixlayout manager to borderpanel **obj** (default layout manager).

j_setflowfill *procedure j_setflowfill (integer obj , integer bool)*
Resizes all containing component to the height (width) of borderpanel **obj**. Needs a flowlayout manager.

j_setflowlayout *procedure j_setflowlayout (integer obj , integer align)*
Adds a flowlayout manager to borderpanel **obj** with the specified **align**ment.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to borderpanel **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setgridlayout *procedure j_setgridlayout (integer obj , integer row , integer col)*
Adds a gridlayout manager to borderpanel **obj** with the specified **row**s and **col**umns.

j_sethgap *procedure j_sethgap (integer obj , integer hgap)*
Sets the horizontal gap between components to **hgap** Pixel.

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

Set the insets to the specified values.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

j_setnolayout *procedure j_setnolayout (integer obj)*
Removes the current layout manager from borderpanel **obj** .

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

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

j_setvgap *procedure j_setvgap (integer obj , integer vgap)*
Sets the vertical gap between components to **hgap** Pixel.

j_sevensegment *integer function j_sevensegment (integer obj , integer color)*
Creates a new sevensegment display with the specified color **color**.

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

j_textarea *integer function j_textarea (integer obj , integer rows , integer columns)*
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

j_textfield *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

Canvas

j_canvas *integer function j_canvas (integer obj , integer width , integer height)*
Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error −1 will be returned.

j_add *procedure j_add (integer obj , integer cont)*
Adds canvas **obj** to container **cont**

j_cliprect *procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)*
Changes current clipping region to the specified rectangle (**x, y, width, height**).

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to canvas **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables canvas **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the canvas **obj**.

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

j_drawcircle *procedure j_drawcircle (integer obj , integer x , integer y , integer r)*
Draws an unfilled circle with center (**x, y**) and radius **x**.

j_drawimage *procedure j_drawimage (integer obj , integer image , integer x , integer y)*
Copies the image, given by its eventnumber **image**, to position (**x, y**).

j_drawimagesource *procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Paints an image at Position (**x, y,**) with **w**idth and **h**eight. The red, green and blue values of each pixel are given by the arrays **r, g, b**.

j_drawline *procedure j_drawline (integer obj , integer x1 , integer y1 , integer x2 , integer y2)*
Draws a line connecting **(x1,y1)** and **(x2,y2)**.

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

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

j_drawpixel *procedure j_drawpixel (integer obj , integer x , integer y)*
Draws a pixel at **(x,y)**.

j_drawpolygon *procedure j_drawpolygon (integer obj , integer len , array of integer x , array of integer y)*
Draws an unfilled polygon based on first **len** elements in **x** and **y**.

j_drawpolyline *procedure j_drawpolyline (integer obj , integer len , array of integer x , array of integer y)*
Draws a series of line segments based on first **len** elements in **x** and **y**.

j_drawrect *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.

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

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

j_drawstring *procedure j_drawstring (integer obj , integer x , integer y , character∗(∗) str)*
Draws text on screen at position (**x,y**).

j_enable *procedure j_enable (integer obj)*
enables the canvas **obj**.

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

j_fillcircle *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
Draws an filled circle with center (**x, y**) and radius **x**.

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

j_fillpolygon *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
Draws an filled polygon based on first **len** elements in **x** and **y**.

j_fillrect

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

j_fillroundrect

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

j_focuslistener

integer function j_focuslistener (integer obj)
Adds a new focus listener to canvas **obj**, and returns its event number.

j_getfontascent

integer function j_getfontascent (integer obj)
Returns the ascent (space above the baseline) of the actual font of canvas **obj**.

j_getfontheight

integer function j_getfontheight (integer obj)
Returns the total pixel height of the actual font of canvas **obj**.

j_getheight

integer function j_getheight (integer obj)
Returns the height of canvas **obj**.

j_getimage

integer function j_getimage (integer obj)
Copy the contents of canvas **obj** into an image and return its eventnumber.

j_getimagesource

integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)
Returns an image of the specified size (**x, y, w**idth, **h**eight) of canvas . The red, green and blue values of each pixel will be stored in **r, g, b**

j_getparentid

integer function j_getparentid (integer obj)
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent

integer function j_getparent (integer obj)
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getscaledimage

integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)
Copy the contents of the rectangular area defined by **x, y,** width **sw**, and height **sh** into an image and return its eventnumber. The image will be scaled to target width **th** and target height **th**.

j_getwidth

integer function j_getwidth (integer obj)
Returns the width of canvas **obj**.

j_getxpos

integer function j_getxpos (integer obj)
Returns the current horizontal position of canvas **obj** in its parent's coordinate space.

j_getypos

integer function j_getypos (integer obj)
Returns the current vertical position of canvas **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the canvas **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to canvas **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to canvas **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the canvas .

j_release *procedure j_release (integer obj)*
Releases canvas **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves canvas **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the canvas 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to canvas **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

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

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

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

j_setxor *procedure j_setxor (integer obj , integer bool)*
Changes painting mode to XOR mode, if bool = .true. . In this mode, drawing the same object in the same color at the same location twice has no net effect.

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

j_translate *procedure j_translate (integer obj , integer x , integer y)*
Moves the origin of drawing operations to (**x, y**).

Checkbox

j_checkbox *integer function j_checkbox (integer obj , character∗(∗) label)*
Creates a new checkbox component with the specified **label** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds checkbox **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to checkbox **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables checkbox **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the checkbox **obj**.

j_enable *procedure j_enable (integer obj)*
enables the checkbox **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to checkbox **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of checkbox **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of checkbox **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of checkbox **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getstate *integer function j_getstate (integer obj)*
Returns .true. , if checkbox is selected, .false. otherwise.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the checkbox 's text or label.

j_getwidth
integer function j_getwidth (integer obj)
Returns the width of checkbox **obj**.

j_getxpos
integer function j_getxpos (integer obj)
Returns the current horizontal position of checkbox **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of checkbox **obj** in its parent's coordinate space.

j_hide
procedure j_hide (integer obj)
Hides the checkbox **obj**.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible
integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
integer function j_keylistener (integer obj)
Adds a new key listener to checkbox **obj**, and returns its event number.

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to checkbox **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu
integer function j_popupmenu (integer obj , character(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print
procedure j_print (integer obj)
prints the checkbox .

j_release
procedure j_release (integer obj)
Releases checkbox **obj** from its parent component (container).

j_setborderpos
procedure j_setborderpos (integer obj , integer pos)
Moves checkbox **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg
procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor
procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcursor
integer function j_setcursor (integer obj , integer cursor)
Changes the checkbox 's **obj** cursor to the specified **cursor**.

j_setfocus
integer function j_setfocus (integer obj)
Directs the input focus to checkbox **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

j_setstate *procedure j_setstate (integer obj , integer bool)*
The checkbox becomes selected, if **bool** is .true. .

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

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

Checkmenuitem

j_checkmenuitem *integer function j_checkmenuitem (integer obj , character*(*) label)*
creates a new checkmenuitem with the specified **label** and returns its event number.

j_disable *procedure j_disable (integer obj)*
Disables checkmenuitem **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the checkmenuitem **obj**.

j_enable *procedure j_enable (integer obj)*
enables the checkmenuitem **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of checkmenuitem 's label or text.

j_getstate *integer function j_getstate (integer obj)*
Returns .true. , if checkmenuitem is selected, .false. otherwise.

j_gettext *procedure j_gettext (integer obj , character*(*) str)*
returns the checkmenuitem 's text or label.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

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

j_setstate *procedure j_setstate (integer obj , integer bool)*
The checkmenuitem becomes selected, if **bool** is .true. .

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

Choice

j_choice *integer function j_choice (integer obj)*
Creates a new choice component and returns its event number.

j_additem *procedure j_additem (integer obj , character∗(∗) str)*
adds a new item containing **str** to choice **obj**.

j_add *procedure j_add (integer obj , integer cont)*
Adds choice **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to choice **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables choice **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the choice **obj**.

j_enable *procedure j_enable (integer obj)*
enables the choice **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to choice **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of choice **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of choice **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of choice **obj**.

j_getitemcount *integer function j_getitemcount (integer obj)*
Returns the number of items of choice **obj**.

j_getitem *procedure j_getitem (integer obj , integer item , character∗(∗) str)*
returns the label of the given **item**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*

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

j_getselect *integer function j_getselect (integer obj)*
Returns the position of currently selected item.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of choice **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of choice **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of choice **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the choice **obj**.

j_insert *integer function j_insert (integer obj , integer pos , character*(*) label)*
inserts a new item to choice **obj** at position **pos** with the specified **label**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to choice **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to choice **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the choice .

j_release *procedure j_release (integer obj)*
Releases choice **obj** from its parent component (container).

j_removeall *integer function j_removeall (integer obj)*
Removes all items from the choice .

j_removeitem *integer function j_removeitem (integer obj , character*(*) item)*
remove the first occurrence of **item** from the choice .

j_remove *integer function j_remove (integer obj , integer item)* removes the Item with the Index **item** from the choice .

j_select *integer function j_select (integer obj , integer item)*
Makes the given **item** the selected one for the choice .

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves choice **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the choice 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to choice **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

Dialog

j_dialog *integer function j_dialog (integer obj , character∗(∗) label)*
Creates a new dialog window with the specified **label** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds dialog **obj** to container **cont**

j_borderpanel *integer function j_borderpanel (integer obj , integer type)*
Creates a new borderpanel component with the style **type** and returns its event number.

j_button *integer function j_button (integer obj , character∗(∗) label)*
Creates a new button component with the specified **label** and returns its event number.

j_canvas *integer function j_canvas (integer obj , integer width , integer height)*
Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error −1 will be returned.

j_checkbox *integer function j_checkbox (integer obj , character∗(∗) label)*
Creates a new checkbox component with the specified **label** and returns its event number.

j_choice *integer function j_choice (integer obj)*
Creates a new choice component and returns its event number.

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to dialog **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables dialog **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the dialog **obj**.

j_enable *procedure j_enable (integer obj)*
enables the dialog **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to dialog **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of dialog **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*

Returns the total pixel height of the actual font of dialog **obj**.

j_getheight
integer function j_getheight (integer obj)
Returns the height of dialog **obj**.

j_getinsets
integer function j_getinsets (integer obj , integer side)
Returns the width of the specified inset.

j_getlayoutid
integer function j_getlayoutid (integer obj)
Returns the event number of the layoutmanager for containers **obj**.

j_getlength
integer function j_getlength (integer obj)
Returns the length of dialog 's label or text.

j_getparentid
integer function j_getparentid (integer obj)
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent
integer function j_getparent (integer obj)
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_gettext
procedure j_gettext (integer obj , character∗(∗) str)
returns the dialog 's text or label.

j_getwidth
integer function j_getwidth (integer obj)
Returns the width of dialog **obj**.

j_getxpos
integer function j_getxpos (integer obj)
Returns the current horizontal position of dialog **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of dialog **obj** in its parent's coordinate space.

j_graphicbutton
integer function j_graphicbutton (integer obj , character∗(∗) filename)
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.

j_graphiclabel
integer function j_graphiclabel (integer obj , character∗(∗) str)
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

j_hide
procedure j_hide (integer obj)
Hides the dialog **obj**.

j_hscrollbar
integer function j_hscrollbar (integer obj)
Creates a new horizontal scrollbar and returns its event number.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isresizable *integer function j_isresizable (integer obj)*
returns true if dialog is resizable, false otherwise

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to dialog **obj**, and returns its event number.

j_label *integer function j_label (integer obj , character∗(∗) label)*
Creates a new label component with the specified **label** and returns its event number.

j_led *integer function j_led (integer obj , integer style , integer color)*
Creates a new led component with the specified **style** and the specified color **color**.

j_line *integer function j_line (integer obj , integer orient , integer style , integer length)*
Creates a new line component with the specified **length** and returns its event number.

j_list *integer function j_list (integer obj , integer rows)*
Creates a new list component with the specified number of **rows** and returns its event number.

j_meter *integer function j_meter (integer obj , character∗(∗) title)*
Creates a new pointer–intrument with the specified label **titel**.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to dialog **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_pack *procedure j_pack (integer obj)*
Resizes dialog to the minimal size of contained components.

j_panel *integer function j_panel (integer obj)*
Creates a new panel component and returns its event number.

j_popupmenu *integer function j_popupmenu (integer obj , character∗(∗) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the dialog .

j_progressbar *integer function j_progressbar (integer obj , integer orient)*
Creates a new progressbar with the specified **orient**ation.

j_radiogroup *integer function j_radiogroup (integer obj)*
Creates a new radiogroup and returns its event number.

j_releaseall *procedure j_releaseall (integer obj)*

Releases all components from dialog **obj**.

j_release *procedure j_release (integer obj)*
Releases dialog **obj** from its parent component (container).

j_scrollpane *integer function j_scrollpane (integer obj)*
Creates a new scrollpane component and returns its event number.

j_setalign *procedure j_setalign (integer obj , integer align)*
Sets the alignment in dialog **obj** to **align**. Needs a flowlayout Manager.

j_setborderlayout *procedure j_setborderlayout (integer obj)*
Adds a borderlayout manager to dialog **obj**.

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves dialog **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the dialog 's **obj** cursor to the specified **cursor**.

j_setfixlayout *procedure j_setfixlayout (integer obj)*
Adds a fixlayout manager to dialog **obj** (default layout manager).

j_setflowfill *procedure j_setflowfill (integer obj , integer bool)*
Resizes all containing component to the height (width) of dialog **obj**. Needs a flowlayout manager.

j_setflowlayout *procedure j_setflowlayout (integer obj , integer align)*
Adds a flowlayout manager to dialog **obj** with the specified **align**ment.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to dialog **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setgridlayout *procedure j_setgridlayout (integer obj , integer row , integer col)*
Adds a gridlayout manager to dialog **obj** with the specified **row**s and **col**umns.

j_sethgap *procedure j_sethgap (integer obj , integer hgap)*
Sets the horizontal gap between components to **hgap** Pixel.

j_setinsets *procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)*
Set the insets to the specified values.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

j_setnolayout *procedure j_setnolayout (integer obj)*
Removes the current layout manager from dialog **obj** .

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

j_setresizable *procedure j_setresizable (integer obj , integer resizable)*
The dialog cannot be resized, if **resizable** is .false. .

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

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

j_setvgap *procedure j_setvgap (integer obj , integer vgap)*
Sets the vertical gap between components to **hgap** Pixel.

j_sevensegment *integer function j_sevensegment (integer obj , integer color)*
Creates a new sevensegment display with the specified color **color**.

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

j_textarea *integer function j_textarea (integer obj , integer rows , integer columns)*
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

j_textfield *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

j_windowlistener *integer function j_windowlistener (integer window , integer kind)*

Adds a new windowlistener to **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

Focuslistener

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to focuslistener **obj**, and returns its event number.

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the focuslistener **obj**.

j_hasfocus *integer function j_hasfocus (integer obj)*
Returns .true. if the focuslistener has the focus, .false. otherwise.

Frame

j_frame *integer function j_frame (character∗(∗) label)*
Creates a new frame component with the specified **label** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds frame **obj** to container **cont**

j_alertbox *procedure j_alertbox (integer obj , character∗(∗) title , character∗(∗) text , character∗(∗) button)*
Shows a alertbox with the specified **title**, **text** and **button**.

j_borderpanel *integer function j_borderpanel (integer obj , integer type)*
Creates a new borderpanel component with the style **type** and returns its event number.

j_button *integer function j_button (integer obj , character∗(∗) label)*
Creates a new button component with the specified **label** and returns its event number.

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

j_checkbox *integer function j_checkbox (integer obj , character∗(∗) label)*
Creates a new checkbox component with the specified **label** and returns its event number.

j_choicebox2 *procedure j_choicebox2 (integer obj , character∗(∗) title , character∗(∗) text , character∗(∗) button1 , character∗(∗) button2)*
Shows a choicebox with the specified **title**, **text** and two buttons.

j_choicebox3 *procedure j_choicebox3 (integer obj , character∗(∗) title , character∗(∗) text , character∗(∗) button1 , character∗(∗) button2 , character∗(∗) button3)*
Shows a choicebox with the specified **title**, **text** and three buttons.

j_choice *integer function j_choice (integer obj)*
Creates a new choice component and returns its event number.

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to frame **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_dialog *integer function j_dialog (integer obj , character∗(∗) label)*
Creates a new dialog window with the specified **label** and returns its event number.

j_disable *procedure j_disable (integer obj)*
Disables frame **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the frame **obj**.

j_enable *procedure j_enable (integer obj)*
enables the frame **obj**.

j_filedialog *procedure j_filedialog (integer frame , character∗(∗) title , character∗(∗) directory , character∗(∗) filename)*
Opens a filedialog box in the specified **directory** with the specified **title** and returns the selected **filename**.

j_fileselector *procedure j_fileselector (integer frame , character∗(∗) title , character∗(∗) filter , character∗(∗) filename)*
Opens a fileslector box with the preselected **filename** and the specified **title** and returns the selected **filename**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to frame **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of frame **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of frame **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of frame **obj**.

j_getinsets *integer function j_getinsets (integer obj , integer side)*
Returns the width of the specified inset.

j_getlayoutid *integer function j_getlayoutid (integer obj)*
Returns the event number of the layoutmanager for containers **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of frame 's label or text.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the frame 's text or label.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of frame **obj**.

j_getxpos
integer function j_getxpos (integer obj)
Returns the current horizontal position of frame **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of frame **obj** in its parent's coordinate space.

j_graphicbutton
integer function j_graphicbutton (integer obj , character(*) filename)*
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.

j_graphiclabel
integer function j_graphiclabel (integer obj , character(*) str)*
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

j_hide
procedure j_hide (integer obj)
Hides the frame **obj**.

j_hscrollbar
integer function j_hscrollbar (integer obj)
Creates a new horizontal scrollbar and returns its event number.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isresizable
integer function j_isresizable (integer obj)
returns true if frame is resizable, false otherwise

j_isvisible
integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
integer function j_keylistener (integer obj)
Adds a new key listener to frame **obj**, and returns its event number.

j_label
integer function j_label (integer obj , character(*) label)*
Creates a new label component with the specified **label** and returns its event number.

j_led
integer function j_led (integer obj , integer style , integer color)
Creates a new led component with the specified **style** and the specified color **color**.

j_line
integer function j_line (integer obj , integer orient , integer style , integer length)
Creates a new line component with the specified **length** and returns its event number.

j_list
integer function j_list (integer obj , integer rows)
Creates a new list component with the specified number of **rows** and returns its event number.

j_menubar
integer function j_menubar (integer obj)
Creates a new menubar and returns its event number.

j_messagebox *procedure j_messagebox (integer obj , character*(*) title , character*(*) text)*
Shows a messagebox with the specified **title** and **text** and returns its event number.

j_meter *integer function j_meter (integer obj , character*(*) title)*
Creates a new pointer–intrument with the specified label **titel**.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to frame **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_pack *procedure j_pack (integer obj)*
Resizes frame to the minimal size of contained components.

j_panel *integer function j_panel (integer obj)*
Creates a new panel component and returns its event number.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_printer *integer function j_printer (integer frame)*
Creates a new object, representing a paper of the printer.

j_print *procedure j_print (integer obj)*
prints the frame .

j_progressbar *integer function j_progressbar (integer obj , integer orient)*
Creates a new progressbar with the specified **orient**ation.

j_radiogroup *integer function j_radiogroup (integer obj)*
Creates a new radiogroup and returns its event number.

j_releaseall *procedure j_releaseall (integer obj)*
Releases all components from frame **obj**.

j_release *procedure j_release (integer obj)*
Releases frame **obj** from its parent component (container).

j_scrollpane *integer function j_scrollpane (integer obj)*
Creates a new scrollpane component and returns its event number.

j_setalign *procedure j_setalign (integer obj , integer align)*
Sets the alignment in frame **obj** to **align**. Needs a flowlayout Manager.

j_setborderlayout *procedure j_setborderlayout (integer obj)*
Adds a borderlayout manager to frame **obj**.

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves frame **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*

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

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the frame 's **obj** cursor to the specified **cursor**.

j_setfixlayout *procedure j_setfixlayout (integer obj)*
Adds a fixlayout manager to frame **obj** (default layout manager).

j_setflowfill *procedure j_setflowfill (integer obj , integer bool)*
Resizes all containing component to the height (width) of frame **obj**. Needs a flowlayout manager.

j_setflowlayout *procedure j_setflowlayout (integer obj , integer align)*
Adds a flowlayout manager to frame **obj** with the specified **align**ment.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to frame **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setgridlayout *procedure j_setgridlayout (integer obj , integer row , integer col)*
Adds a gridlayout manager to frame **obj** with the specified **row**s and **col**umns.

j_sethgap *procedure j_sethgap (integer obj , integer hgap)*
Sets the horizontal gap between components to **hgap** Pixel.

j_seticon *procedure j_seticon (integer frame , integer icon)*
Sets the image **icon** to display when the **frame** is iconized. Not all platforms support the concept of iconizing a window.

j_setinsets *procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)*
Set the insets to the specified values.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

j_setnolayout *procedure j_setnolayout (integer obj)*
Removes the current layout manager from frame **obj** .

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

j_setresizable *procedure j_setresizable (integer obj , integer resizable)*
The frame cannot be resized, if **resizable** is .false. .

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

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

j_setvgap *procedure j_setvgap (integer obj , integer vgap)*
Sets the vertical gap between components to **hgap** Pixel.

j_sevensegment *integer function j_sevensegment (integer obj , integer color)*
Creates a new sevensegment display with the specified color **color**.

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

j_textarea *integer function j_textarea (integer obj , integer rows , integer columns)*
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

j_textfield *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

j_windowlistener *integer function j_windowlistener (integer window , integer kind)*
Adds a new windowlistener to **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_window *integer function j_window (integer obj)*
Creates a new simple window and returns its event number.

Helpmenu

j_helpmenu *integer function j_helpmenu (integer obj , character∗(∗) label)*
Creates a new helpmenu component with the specified **label** and returns its event number.

j_checkmenuitem *integer function j_checkmenuitem (integer obj , character∗(∗) label)*
creates a new checkmenuitem with the specified **label** and returns its event number.

j_disable *procedure j_disable (integer obj)*
Disables helpmenu **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the helpmenu **obj**.

j_enable *procedure j_enable (integer obj)*
enables the helpmenu **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of helpmenu 's label or text.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the helpmenu 's text or label.

j_menuitem *integer function j_menuitem (integer obj , character∗(∗) label)*
Creates a new menuitem with the specified **label** and returns its event number.

j_seperator *procedure j_seperator (integer obj)*
Adds a separator bar to the helpmenu .

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

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

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

Hscrollbar

j_hscrollbar *integer function j_hscrollbar (integer obj)*
Creates a new horizontal scrollbar and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds hscrollbar **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to hscrollbar **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables hscrollbar **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the hscrollbar **obj**.

j_enable *procedure j_enable (integer obj)*
enables the hscrollbar **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to hscrollbar **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of hscrollbar **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of hscrollbar **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of hscrollbar **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getvalue *integer function j_getvalue (integer obj)*
Returns the current setting of the scrollbar.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of hscrollbar **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of hscrollbar **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of hscrollbar **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the hscrollbar **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to hscrollbar **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to hscrollbar **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu _integer function j_popupmenu (integer obj , character*(*) label)_
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the hscrollbar .

j_release *procedure j_release (integer obj)*
Releases hscrollbar **obj** from its parent component (container).

j_setblockinc *integer function j_setblockinc (integer obj , integer val)*
Changes the block increment amount for the hscrollbar to **val**.

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves hscrollbar **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the hscrollbar 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to hscrollbar **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setmax *integer function j_setmax (integer obj , integer val)*
Changes the maximum value for the hscrollbar to **val**.

j_setmin *integer function j_setmin (integer obj , integer val)*
Changes the minimum value for the hscrollbar to **val**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

j_setslidesize *integer function j_setslidesize (integer obj , integer val)*
Changes the slide size to **val**.

j_setunitinc *integer function j_setunitinc (integer obj , integer val)*
Changes the unit increment amount for the hscrollbar to **val**

j_setvalue *procedure j_setvalue (integer obj , integer val)*
Changes the current value of the hscrollbar to **val**.

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

Graphicbutton

j_graphicbutton *integer function j_graphicbutton (integer obj , character∗(∗) filename)*
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds graphicbutton **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to graphicbutton **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables graphicbutton **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the graphicbutton **obj**.

j_enable *procedure j_enable (integer obj)*
enables the graphicbutton **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to graphicbutton **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of graphicbutton **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of graphicbutton **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of graphicbutton **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of graphicbutton **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of graphicbutton **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of graphicbutton **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the graphicbutton **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to graphicbutton **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to graphicbutton **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the graphicbutton .

j_release *procedure j_release (integer obj)*
Releases graphicbutton **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves graphicbutton **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the graphicbutton 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to graphicbutton **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setimage *procedure j_setimage (integer obj , integer image)*
Sets the **image** to be displayed in **obj**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

Graphiclabel

j_graphiclabel *integer function j_graphiclabel (integer obj , character*(*) str)*
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds graphiclabel **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to graphiclabel **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables graphiclabel **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the graphiclabel **obj**.

j_enable *procedure j_enable (integer obj)*
enables the graphiclabel **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to graphiclabel **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of graphiclabel **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of graphiclabel **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of graphiclabel **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of graphiclabel **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of graphiclabel **obj** in its parent's coordinate space.

j_getypos

integer function j_getypos (integer obj)
Returns the current vertical position of graphiclabel **obj** in its parent's coordinate space.

j_hide

procedure j_hide (integer obj)
Hides the graphiclabel **obj**.

j_isparent

integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible

integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener

integer function j_keylistener (integer obj)
Adds a new key listener to graphiclabel **obj**, and returns its event number.

j_mouselistener

integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to graphiclabel **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu

integer function j_popupmenu (integer obj , character(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print

procedure j_print (integer obj)
prints the graphiclabel .

j_release

procedure j_release (integer obj)
Releases graphiclabel **obj** from its parent component (container).

j_setborderpos

procedure j_setborderpos (integer obj , integer pos)
Moves graphiclabel **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg

procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor

procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcursor

integer function j_setcursor (integer obj , integer cursor)
Changes the graphiclabel 's **obj** cursor to the specified **cursor**.

j_setfocus

integer function j_setfocus (integer obj)
Directs the input focus to graphiclabel **obj**.

j_setfontname

procedure j_setfontname (integer obj , integer name)
Changes the font to the given **name**.

j_setfont

procedure j_setfont (integer obj , integer name , integer style , integer size)
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setimage *procedure j_setimage (integer obj , integer image)*
Sets the **image** to be displayed in **obj**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

Image

j_image
integer function j_image (integer width , integer height)
Creates a new (memory) image component with the given **width** and **height** and returns its event number.

j_cliprect
procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)
Changes current clipping region to the specified rectangle (**x, y, width, height**).

j_dispose
procedure j_dispose (integer obj)
Releases the resources of the image **obj**.

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

j_drawcircle
procedure j_drawcircle (integer obj , integer x , integer y , integer r)
Draws an unfilled circle with center (**x, y**) and radius **x**.

j_drawimage
procedure j_drawimage (integer obj , integer image , integer x , integer y)
Copies the image, given by its eventnumber **image**, to position (**x, y**).

j_drawimagesource
procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)
Paints an image at Position (**x, y,**) with **w**idth and **h**eight. The red, green and blue values of each pixel are given by the arrays **r, g, b**.

j_drawline
procedure j_drawline (integer obj , integer x1 , integer y1 , integer x2 , integer y2)
Draws a line connecting **(x1,y1)** and **(x2,y2)**.

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

j_drawpixel
procedure j_drawpixel (integer obj , integer x , integer y)
Draws a pixel at **(x,y)**.

j_drawpolygon
procedure j_drawpolygon (integer obj , integer len , array of integer x , array of integer y)
Draws an unfilled polygon based on first **len** elements in **x** and **y**.

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

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

j_drawrect *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.

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

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

j_drawstring *procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)*
Draws text on screen at position (**x,y**).

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

j_fillcircle *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
Draws an filled circle with center (**x, y**) and radius **x**.

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

j_fillpolygon *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
Draws an filled polygon based on first **len** elements in **x** and **y**.

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

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

j_getheight *integer function j_getheight (integer obj)*
Returns the height of image **obj**.

j_getimage *integer function j_getimage (integer obj)*
Copy the contents of image **obj** into an image and return its eventnumber.

j_getimagesource *integer function j_getimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Returns an image of the specified size (**x, y, w**idth, **h**eight) of image . The red, green and blue values of each pixel will be stored in **r, g, b**

j_getscaledimage *integer function j_getscaledimage (integer obj , integer x , integer y , integer sw , integer sh , integer tw , integer th)*
Copy the contents of the rectangular area defined by **x, y,** width **sw**, and height **sh** into an image and return its eventnumber. The image will be scaled to target width **th** and target height **th**.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of image **obj**.

j_print *procedure j_print (integer obj)*
prints the image .

j_setxor *procedure j_setxor (integer obj , integer bool)*
Changes painting mode to XOR mode, if bool = .true. . In this mode, drawing the same object in the same color at the same location twice has no net effect.

j_translate *procedure j_translate (integer obj , integer x , integer y)*
Moves the origin of drawing operations to (**x, y**).

Keylistener

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to keylistener **obj**, and returns its event number.

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the keylistener **obj**.

j_getkeychar *integer function j_getkeychar (integer obj)*
Returns the ascii value of the last pressed key.

j_getkeycode *integer function j_getkeycode (integer obj)*
Returns the integer key code of the last pressed key.

Label

j_label *integer function j_label (integer obj , character∗(∗) label)*
Creates a new label component with the specified **label** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds label **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to label **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables label **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the label **obj**.

j_enable *procedure j_enable (integer obj)*
enables the label **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to label **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of label **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of label **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of label **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the label 's text or label.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of label **obj**.

j_getxpos *integer function j_getxpos (integer obj)*

Returns the current horizontal position of label **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of label **obj** in its parent's coordinate space.

j_hide
procedure j_hide (integer obj)
Hides the label **obj**.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible
integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
integer function j_keylistener (integer obj)
Adds a new key listener to label **obj**, and returns its event number.

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to label **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu
integer function j_popupmenu (integer obj , character(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print
procedure j_print (integer obj)
prints the label .

j_release
procedure j_release (integer obj)
Releases label **obj** from its parent component (container).

j_setborderpos
procedure j_setborderpos (integer obj , integer pos)
Moves label **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg
procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor
procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcursor
integer function j_setcursor (integer obj , integer cursor)
Changes the label 's **obj** cursor to the specified **cursor**.

j_setfocus
integer function j_setfocus (integer obj)
Directs the input focus to label **obj**.

j_setfontname
procedure j_setfontname (integer obj , integer name)
Changes the font to the given **name**.

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

Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

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

Led

j_led *integer function j_led (integer obj , integer style , integer color)*
Creates a new led component with the specified **style** and the specified color **color**.

j_add *procedure j_add (integer obj , integer cont)*
Adds led **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to led **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables led **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the led **obj**.

j_enable *procedure j_enable (integer obj)*
enables the led **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to led **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of led **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of led **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of led **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getstate *integer function j_getstate (integer obj)*
Returns .true. , if led is selected, .false. otherwise.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of led **obj**.

j_getxpos *integer function j_getxpos (integer obj)*

Returns the current horizontal position of led **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of led **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the led **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to led **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to led **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the led .

j_release *procedure j_release (integer obj)*
Releases led **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves led **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the led 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to led **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

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

Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

j_setstate *procedure j_setstate (integer obj , integer bool)*
The led becomes selected, if **bool** is .true. .

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

List

j_list *integer function j_list (integer obj , integer rows)*
Creates a new list component with the specified number of **rows** and returns its event number.

j_additem *procedure j_additem (integer obj , character*(*) str)*
adds a new item containing **str** to list **obj**.

j_add *procedure j_add (integer obj , integer cont)*
Adds list **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to list **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_deselect *integer function j_deselect (integer obj , integer item)*
Deselects the item at the designated position **item**, if selected.

j_disable *procedure j_disable (integer obj)*
Disables list **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the list **obj**.

j_enable *procedure j_enable (integer obj)*
enables the list **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to list **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of list **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of list **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of list **obj**.

j_getitemcount *integer function j_getitemcount (integer obj)*
Returns the number of items of list **obj**.

j_getitem *procedure j_getitem (integer obj , integer item , character*(*) str)*
returns the label of the given **item**.

j_getparentid *integer function j_getparentid (integer obj)*

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

j_getparent
integer function j_getparent (integer obj)
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getselect
integer function j_getselect (integer obj)
Returns the position of currently selected item.

j_getwidth
integer function j_getwidth (integer obj)
Returns the width of list **obj**.

j_getxpos
integer function j_getxpos (integer obj)
Returns the current horizontal position of list **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of list **obj** in its parent's coordinate space.

j_hide
procedure j_hide (integer obj)
Hides the list **obj**.

j_insert
integer function j_insert (integer obj , integer pos , character(*) label)*
inserts a new item to list **obj** at position **pos** with the specified **label**.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isselect
integer function j_isselect (integer obj , integer item)
Returns .true. if the particular **item** is currently selected, .false. otherwise.

j_isvisible
integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
integer function j_keylistener (integer obj)
Adds a new key listener to list **obj**, and returns its event number.

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to list **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_multiplemode
integer function j_multiplemode (integer obj , integer bool)
if **bool** is .true. , selection mode is turned to multiplemode.

j_popupmenu
integer function j_popupmenu (integer obj , character(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print
procedure j_print (integer obj)
prints the list .

j_release
procedure j_release (integer obj)

Releases list **obj** from its parent component (container).

j_removeall *integer function j_removeall (integer obj)*
Removes all items from the list .

j_removeitem *integer function j_removeitem (integer obj , character*(*) item)*
remove the first occurrence of **item** from the list .

j_remove *integer function j_remove (integer obj , integer item)*
removes the Item with the Index **item** from the list .

j_select *integer function j_select (integer obj , integer item)*
Makes the given **item** the selected one for the list .

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves list **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the list 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to list **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

Resizes list **obj** to specified **width** and **height**.

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

Menu

j_menu *integer function j_menu (integer obj , character∗(∗) str)*
Creates a new menu component with the specified **label** and returns its event number.

j_checkmenuitem *integer function j_checkmenuitem (integer obj , character∗(∗) label)*
creates a new checkmenuitem with the specified **label** and returns its event number.

j_disable *procedure j_disable (integer obj)*
Disables menu **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the menu **obj**.

j_enable *procedure j_enable (integer obj)*
enables the menu **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of menu 's label or text.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the menu 's text or label.

j_helpmenu *integer function j_helpmenu (integer obj , character∗(∗) label)*
Creates a new helpmenu component with the specified **label** and returns its event number.

j_menuitem *integer function j_menuitem (integer obj , character∗(∗) label)*
Creates a new menuitem with the specified **label** and returns its event number.

j_menu *integer function j_menu (integer obj , character∗(∗) str)*
Creates a new menu component with the specified **label** and returns its event number.

j_seperator *procedure j_seperator (integer obj)*
Adds a separator bar to the menu .

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

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

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

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

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

Menuitem

j_menuitem *integer function j_menuitem (integer obj , character∗(∗) label)*
Creates a new menuitem with the specified **label** and returns its event number.

j_disable *procedure j_disable (integer obj)*
Disables menuitem **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the menuitem **obj**.

j_enable *procedure j_enable (integer obj)*
enables the menuitem **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of menuitem 's label or text.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the menuitem 's text or label.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

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

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

Meter

j_meter *integer function j_meter (integer obj , character∗(∗) title)*
Creates a new pointer–intrument with the specified label **titel**.

j_add *procedure j_add (integer obj , integer cont)*
Adds meter **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to meter **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables meter **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the meter **obj**.

j_enable *procedure j_enable (integer obj)*
enables the meter **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to meter **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of meter **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of meter **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of meter **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of meter **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of meter **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of meter **obj** in its parent's coordinate space.

j_hide
procedure j_hide (integer obj)
Hides the meter **obj**.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible
integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
integer function j_keylistener (integer obj)
Adds a new key listener to meter **obj**, and returns its event number.

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to meter **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu
integer function j_popupmenu (integer obj , character(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print
procedure j_print (integer obj)
prints the meter .

j_release
procedure j_release (integer obj)
Releases meter **obj** from its parent component (container).

j_setborderpos
procedure j_setborderpos (integer obj , integer pos)
Moves meter **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg
procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor
procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcursor
integer function j_setcursor (integer obj , integer cursor)
Changes the meter 's **obj** cursor to the specified **cursor**.

j_setfocus
integer function j_setfocus (integer obj)
Directs the input focus to meter **obj**.

j_setfontname
procedure j_setfontname (integer obj , integer name)
Changes the font to the given **name**.

j_setfont
procedure j_setfont (integer obj , integer name , integer style , integer size)
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setmax *integer function j_setmax (integer obj , integer val)*
Changes the maximum value for the meter to **val**.

j_setmin *integer function j_setmin (integer obj , integer val)*
Changes the minimum value for the meter to **val**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

j_setvalue *procedure j_setvalue (integer obj , integer val)*
Changes the current value of the meter to **val**.

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

Mouselistener

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to mouselistener **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_dispose
procedure j_dispose (integer obj)
Releases the resources of the mouselistener **obj**.

j_getmousebutton
integer function j_getmousebutton (integer mouselistener)
Returns the latest used mousebutton.

j_getmousex
integer function j_getmousex (integer mouselistener)
Returns the current horizontal position of the mouse in its parent's coordinate space.

j_getmousey
integer function j_getmousey (integer mouselistener)
Returns the current vertical position of the mouse in its parent's coordinate space.

Panel

j_panel *integer function j_panel (integer obj)*
Creates a new panel component and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds panel **obj** to container **cont**

j_borderpanel *integer function j_borderpanel (integer obj , integer type)*
Creates a new borderpanel component with the style **type** and returns its event number.

j_button *integer function j_button (integer obj , character*(*) label)*
Creates a new button component with the specified **label** and returns its event number.

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

j_checkbox *integer function j_checkbox (integer obj , character*(*) label)*
Creates a new checkbox component with the specified **label** and returns its event number.

j_choice *integer function j_choice (integer obj)*
Creates a new choice component and returns its event number.

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to panel **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables panel **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the panel **obj**.

j_enable *procedure j_enable (integer obj)*
enables the panel **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to panel **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of panel **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*

Returns the total pixel height of the actual font of panel **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of panel **obj**.

j_getinsets *integer function j_getinsets (integer obj , integer side)*
Returns the width of the specified inset.

j_getlayoutid *integer function j_getlayoutid (integer obj)*
Returns the event number of the layoutmanager for containers **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of panel **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of panel **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of panel **obj** in its parent's coordinate space.

j_graphicbutton *integer function j_graphicbutton (integer obj , character∗(∗) filename)*
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.

j_graphiclabel *integer function j_graphiclabel (integer obj , character∗(∗) str)*
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

j_hide *procedure j_hide (integer obj)*
Hides the panel **obj**.

j_hscrollbar *integer function j_hscrollbar (integer obj)*
Creates a new horizontal scrollbar and returns its event number.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to panel **obj**, and returns its event number.

j_label *integer function j_label (integer obj , character∗(∗) label)*
Creates a new label component with the specified **label** and returns its event number.

j_led *integer function j_led (integer obj , integer style , integer color)*
Creates a new led component with the specified **style** and the specified color **color**.

j_line *integer function j_line (integer obj , integer orient , integer style , integer length)*
Creates a new line component with the specified **length** and returns its event number.

j_list *integer function j_list (integer obj , integer rows)*
Creates a new list component with the specified number of **rows** and returns its event number.

j_meter *integer function j_meter (integer obj , character∗(∗) title)*
Creates a new pointer–intrument with the specified label **titel**.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to panel **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_pack *procedure j_pack (integer obj)*
Resizes panel to the minimal size of contained components.

j_panel *integer function j_panel (integer obj)*
Creates a new panel component and returns its event number.

j_popupmenu *integer function j_popupmenu (integer obj , character∗(∗) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the panel .

j_progressbar *integer function j_progressbar (integer obj , integer orient)*
Creates a new progressbar with the specified **orient**ation.

j_radiogroup *integer function j_radiogroup (integer obj)*
Creates a new radiogroup and returns its event number.

j_releaseall *procedure j_releaseall (integer obj)*
Releases all components from panel **obj**.

j_release *procedure j_release (integer obj)*
Releases panel **obj** from its parent component (container).

j_scrollpane *integer function j_scrollpane (integer obj)*
Creates a new scrollpane component and returns its event number.

j_setalign *procedure j_setalign (integer obj , integer align)*

Sets the alignment in panel **obj** to **align**. Needs a flowlayout Manager.

j_setborderlayout
procedure j_setborderlayout (integer obj)
Adds a borderlayout manager to panel **obj**.

j_setborderpos
procedure j_setborderpos (integer obj , integer pos)
Moves panel **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg
procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor
procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcursor
integer function j_setcursor (integer obj , integer cursor)
Changes the panel 's **obj** cursor to the specified **cursor**.

j_setfixlayout
procedure j_setfixlayout (integer obj)
Adds a fixlayout manager to panel **obj** (default layout manager).

j_setflowfill
procedure j_setflowfill (integer obj , integer bool)
Resizes all containing component to the height (width) of panel **obj**. Needs a flowlayout manager.

j_setflowlayout
procedure j_setflowlayout (integer obj , integer align)
Adds a flowlayout manager to panel **obj** with the specified **align**ment.

j_setfocus
integer function j_setfocus (integer obj)
Directs the input focus to panel **obj**.

j_setfontname
procedure j_setfontname (integer obj , integer name)
Changes the font to the given **name**.

j_setfont
procedure j_setfont (integer obj , integer name , integer style , integer size)
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize
procedure j_setfontsize (integer obj , integer size)
Changes the font to the given **size**.

j_setfontstyle
procedure j_setfontstyle (integer obj , integer style)
Changes the font to the given **style**.

j_setgridlayout
procedure j_setgridlayout (integer obj , integer row , integer col)
Adds a gridlayout manager to panel **obj** with the specified **row**s and **col**umns.

j_sethgap
procedure j_sethgap (integer obj , integer hgap)
Sets the horizontal gap between components to **hgap** Pixel.

j_setinsets
procedure j_setinsets (integer obj , integer top , integer bottom , integer left , integer right)
Set the insets to the specified values.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

j_setnolayout *procedure j_setnolayout (integer obj)*
Removes the current layout manager from panel **obj** .

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

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

j_setvgap *procedure j_setvgap (integer obj , integer vgap)*
Sets the vertical gap between components to **hgap** Pixel.

j_sevensegment *integer function j_sevensegment (integer obj , integer color)*
Creates a new sevensegment display with the specified color **color**.

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

j_textarea *integer function j_textarea (integer obj , integer rows , integer columns)*
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

j_textfield *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

Popupmenu

j_popupmenu *integer function j_popupmenu (integer obj , character∗(∗) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_checkmenuitem *integer function j_checkmenuitem (integer obj , character∗(∗) label)*
creates a new checkmenuitem with the specified **label** and returns its event number.

j_disable *procedure j_disable (integer obj)*
Disables popupmenu **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the popupmenu **obj**.

j_enable *procedure j_enable (integer obj)*
enables the popupmenu **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of popupmenu 's label or text.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the popupmenu 's text or label.

j_menuitem *integer function j_menuitem (integer obj , character∗(∗) label)*
Creates a new menuitem with the specified **label** and returns its event number.

j_seperator *procedure j_seperator (integer obj)*
Adds a separator bar to the popupmenu .

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

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

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

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

Printer

j_printer *integer function j_printer (integer frame)*
Creates a new object, representing a paper of the printer.

j_cliprect *procedure j_cliprect (integer obj , integer x , integer y , integer width , integer height)*
Changes current clipping region to the specified rectangle (**x, y, width, height**).

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the printer **obj**.

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

j_drawcircle *procedure j_drawcircle (integer obj , integer x , integer y , integer r)*
Draws an unfilled circle with center (**x, y**) and radius **x**.

j_drawimage *procedure j_drawimage (integer obj , integer image , integer x , integer y)*
Copies the image, given by its eventnumber **image**, to position (**x, y**).

j_drawimagesource *procedure j_drawimagesource (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)*
Paints an image at Position (**x, y,**) with **w**idth and **h**eight. The red, green and blue values of each pixel are given by the arrays **r, g, b**.

j_drawline *procedure j_drawline (integer obj , integer x1 , integer y1 , integer x2 , integer y2)*
Draws a line connecting **(x1,y1)** and **(x2,y2)**.

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

j_drawpixel *procedure j_drawpixel (integer obj , integer x , integer y)*
Draws a pixel at **(x,y)**.

j_drawpolygon *procedure j_drawpolygon (integer obj , integer len , array of integer x , array of integer y)*
Draws an unfilled polygon based on first **len** elements in **x** and **y**.

j_drawpolyline *procedure j_drawpolyline (integer obj , integer len , array of integer x , array of integer y)*
Draws a series of line segments based on first **len** elements in **x** and **y**.

j_drawrect *procedure j_drawrect (integer obj , integer x , integer y , integer width , integer height)*
Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.

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

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

j_drawstring *procedure j_drawstring (integer obj , integer x , integer y , character*(*) str)*
Draws text on screen at position (**x,y**).

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

j_fillcircle *procedure j_fillcircle (integer obj , integer x , integer y , integer r)*
Draws an filled circle with center (**x, y**) and radius **x**.

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

j_fillpolygon *procedure j_fillpolygon (integer obj , integer len , array of integer x , array of integer y)*
Draws an filled polygon based on first **len** elements in **x** and **y**.

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

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

j_print *procedure j_print (integer obj)*
prints the printer .

j_setxor *procedure j_setxor (integer obj , integer bool)*
Changes painting mode to XOR mode, if bool = .true. . In this mode, drawing the same object in the same color at the same location twice has no net effect.

j_translate *procedure j_translate (integer obj , integer x , integer y)*
Moves the origin of drawing operations to (**x, y**).

Progressbar

j_progressbar *integer function j_progressbar (integer obj , integer orient)*
Creates a new progressbar with the specified **orient**ation.

j_add *procedure j_add (integer obj , integer cont)*
Adds progressbar **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to progressbar **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables progressbar **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the progressbar **obj**.

j_enable *procedure j_enable (integer obj)*
enables the progressbar **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to progressbar **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of progressbar **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of progressbar **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of progressbar **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of progressbar **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of progressbar **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of progressbar **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the progressbar **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to progressbar **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to progressbar **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the progressbar .

j_release *procedure j_release (integer obj)*
Releases progressbar **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves progressbar **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the progressbar 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to progressbar **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

Radiobutton

j_radiobutton *integer function j_radiobutton (integer obj , character∗(∗) label)*
Creates a new radiobutton with the specified **label** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds radiobutton **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to radiobutton **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables radiobutton **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the radiobutton **obj**.

j_enable *procedure j_enable (integer obj)*
enables the radiobutton **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to radiobutton **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of radiobutton **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of radiobutton **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of radiobutton **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getstate *integer function j_getstate (integer obj)*
Returns .true. , if radiobutton is selected, .false. otherwise.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the radiobutton 's text or label.

j_getwidth
integer function j_getwidth (integer obj)
Returns the width of radiobutton **obj**.

j_getxpos
integer function j_getxpos (integer obj)
Returns the current horizontal position of radiobutton **obj** in its parent's coordinate space.

j_getypos
integer function j_getypos (integer obj)
Returns the current vertical position of radiobutton **obj** in its parent's coordinate space.

j_hide
procedure j_hide (integer obj)
Hides the radiobutton **obj**.

j_isparent
integer function j_isparent (integer obj , integer cont)
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible
integer function j_isvisible (integer obj)
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener
integer function j_keylistener (integer obj)
Adds a new key listener to radiobutton **obj**, and returns its event number.

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to radiobutton **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu
integer function j_popupmenu (integer obj , character∗(∗) label)
Creates a new popupmenu with the specified **label** and returns its event number.

j_print
procedure j_print (integer obj)
prints the radiobutton .

j_release
procedure j_release (integer obj)
Releases radiobutton **obj** from its parent component (container).

j_setborderpos
procedure j_setborderpos (integer obj , integer pos)
Moves radiobutton **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg
procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor
procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcursor
integer function j_setcursor (integer obj , integer cursor)
Changes the radiobutton 's **obj** cursor to the specified **cursor**.

j_setfocus
integer function j_setfocus (integer obj)
Directs the input focus to radiobutton **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

j_setradiogroup *integer function j_setradiogroup (integer rbutton, , integer rgroup)*
Sets radiobuttons **rbutton** group to be the specified radiogroup **rgroup**. If the radiobuttons is already in a different radiogroup, it is first taken out of that group.

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

j_setstate *procedure j_setstate (integer obj , integer bool)*
The radiobutton becomes selected, if **bool** is .true. .

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

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

Sevensegment

j_sevensegment *integer function j_sevensegment (integer obj , integer color)*
Creates a new sevensegment display with the specified color **color**.

j_add *procedure j_add (integer obj , integer cont)*
Adds sevensegment–component **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to sevensegment–component **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables sevensegment–component **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the sevensegment–component **obj**.

j_enable *procedure j_enable (integer obj)*
enables the sevensegment–component **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to sevensegment–component **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of sevensegment–component **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of sevensegment–component **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of sevensegment–component **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of sevensegment–component **obj**.

j_getxpos *integer function j_getxpos (integer obj)*

Returns the current horizontal position of sevensegment–component **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of sevensegment–component **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the sevensegment–component **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to sevensegment–component **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to sevensegment–component **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the sevensegment–component .

j_release *procedure j_release (integer obj)*
Releases sevensegment–component **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves sevensegment–component **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the sevensegment–component 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to sevensegment–component **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

j_setpos *procedure j_setpos (integer obj , integer xpos , integer ypos)*
Relocates the sevensegment–component **obj** to the specified Position (**xpos**,**ypos**).

j_setsize *procedure j_setsize (integer obj , integer width , integer height)*
Resizes sevensegment–component **obj** to specified **width** and **height**.

j_setvalue *procedure j_setvalue (integer obj , integer val)*
Changes the current value of the sevensegment–component to **val**.

j_show *procedure j_show (integer obj)*
Shows the sevensegment–component **obj**.

Scrollpane

j_scrollpane *integer function j_scrollpane (integer obj)*
Creates a new scrollpane component and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds scrollpane **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to scrollpane **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables scrollpane **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the scrollpane **obj**.

j_enable *procedure j_enable (integer obj)*
enables the scrollpane **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to scrollpane **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of scrollpane **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of scrollpane **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of scrollpane **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getviewportheight *integer function j_getviewportheight (integer obj)*
Returns the height of the scrollpane 's **obj** port (the area that is shown)

j_getviewportwidth *integer function j_getviewportwidth (integer obj)*
Returns the width of the scrollpane 's **obj** port (the area that is shown)

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of scrollpane **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of scrollpane **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of scrollpane **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the scrollpane **obj**.

j_hscrollbar *integer function j_hscrollbar (integer obj)*
Creates a new horizontal scrollbar and returns its event number.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to scrollpane **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to scrollpane **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character∗(∗) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the scrollpane .

j_release *procedure j_release (integer obj)*
Releases scrollpane **obj** from its parent component (container).

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves scrollpane **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the scrollpane 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to scrollpane **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

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

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

Textarea

j_textarea *integer function j_textarea (integer obj , integer rows , integer columns)*
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds textarea **obj** to container **cont**

j_appendtext *procedure j_appendtext (integer obj , character*(*) text)*
Appends the given **text** to the **obj** current text.

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to textarea **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_delete *procedure j_delete (integer obj , integer start , integer end)*
Delets text from starting position **start** to ending position **end**.

j_disable *procedure j_disable (integer obj)*
Disables textarea **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the textarea **obj**.

j_enable *procedure j_enable (integer obj)*
enables the textarea **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to textarea **obj**, and returns its event number.

j_getcolumns *procedure j_getcolumns (integer obj)*
Gets the number of columns in **obj**.

j_getcurpos *integer function j_getcurpos (integer obj)*
Returns the position, in characters, of the text cursor.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of textarea **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of textarea **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of textarea **obj**.

j_getlength *integer function j_getlength (integer obj)*

Returns the length of textarea 's label or text.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getrows *procedure j_getrows (integer obj)*
Gets the number of rows in **obj**.

j_getselend *integer function j_getselend (integer obj)*
Returns the ending position of any selected text.

j_getselstart *integer function j_getselstart (integer obj)*
Returns the initial position of any selected text.

j_getseltext *procedure j_getseltext (integer obj , character*(*) text)*
Returns the currently selected text of textarea **obj**.

j_gettext *procedure j_gettext (integer obj , character*(*) str)*
returns the textarea 's text or label.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of textarea **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of textarea **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of textarea **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the textarea **obj**.

j_inserttext *procedure j_inserttext (integer obj , character*(*) text , integer pos)*
Places additional text within the textarea at the given position **pos**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to textarea **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*

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

j_popupmenu

integer function j_popupmenu (integer obj , character(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print

procedure j_print (integer obj)
prints the textarea .

j_release

procedure j_release (integer obj)
Releases textarea **obj** from its parent component (container).

j_replacetext

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

j_selectall

procedure j_selectall (integer obj)
Selects all the text in the textarea .

j_selecttext

procedure j_selecttext (integer obj , integer start , integer end)
Selects text from starting position **start** to ending position **end**.

j_setborderpos

procedure j_setborderpos (integer obj , integer pos)
Moves textarea **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg

procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor

procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcolumns

procedure j_setcolumns (integer obj , integer columns)
Sets the number of columns for **obj** to **columns**.

j_setcurpos

procedure j_setcurpos (integer obj , integer pos)
Change the location of the text cursor to the specified position **pos**.

j_setcursor

integer function j_setcursor (integer obj , integer cursor)
Changes the textarea 's **obj** cursor to the specified **cursor**.

j_seteditable

procedure j_seteditable (integer obj , integer bool)
Allows to make the textarea editable (**bool**=.true.) or read-only (**bool**=.false.).

j_setfocus

integer function j_setfocus (integer obj)
Directs the input focus to textarea **obj**.

j_setfontname

procedure j_setfontname (integer obj , integer name)
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

j_setrows *procedure j_setrows (integer obj , integer rows)*
Sets the number of rows for **obj** to **rows**.

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

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

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

Textfield

j_textfield *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds textfield **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to textfield **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables textfield **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the textfield **obj**.

j_enable *procedure j_enable (integer obj)*
enables the textfield **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to textfield **obj**, and returns its event number.

j_getcolumns *procedure j_getcolumns (integer obj)*
Gets the number of columns in **obj**.

j_getcurpos *integer function j_getcurpos (integer obj)*
Returns the position, in characters, of the text cursor.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of textfield **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of textfield **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of textfield **obj**.

j_getlength *integer function j_getlength (integer obj)*
Returns the length of textfield 's label or text.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame -1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getselend *integer function j_getselend (integer obj)*
Returns the ending position of any selected text.

j_getselstart *integer function j_getselstart (integer obj)*
Returns the initial position of any selected text.

j_getseltext *procedure j_getseltext (integer obj , character∗(∗) text)*
Returns the currently selected text of textfield **obj**.

j_gettext *procedure j_gettext (integer obj , character∗(∗) str)*
returns the textfield 's text or label.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of textfield **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of textfield **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of textfield **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the textfield **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to textfield **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to textfield **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character∗(∗) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the textfield .

j_release *procedure j_release (integer obj)*
Releases textfield **obj** from its parent component (container).

j_selectall
procedure j_selectall (integer obj)
Selects all the text in the textfield .

j_selecttext
procedure j_selecttext (integer obj , integer start , integer end)
Selects text from starting position **start** to ending position **end**.

j_setborderpos
procedure j_setborderpos (integer obj , integer pos)
Moves textfield **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg
procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)
Sets the background color to the (**r, g, b**) values.

j_setcolor
procedure j_setcolor (integer obj , integer r , integer g, , integer b)
Sets the foreground color to the (**r, g, b**) values.

j_setcolumns
procedure j_setcolumns (integer obj , integer columns)
Sets the number of columns for **obj** to **columns**.

j_setcurpos
procedure j_setcurpos (integer obj , integer pos)
Change the location of the text cursor to the specified position **pos**.

j_setcursor
integer function j_setcursor (integer obj , integer cursor)
Changes the textfield 's **obj** cursor to the specified **cursor**.

j_setechochar
procedure j_setechochar (integer obj , character chr)
Changes the character **chr** that is used to echo all user input in the textfield .

j_seteditable
procedure j_seteditable (integer obj , integer bool)
Allows to make the textfield editable (**bool**=.true.) or read-only (**bool**=.false.).

j_setfocus
integer function j_setfocus (integer obj)
Directs the input focus to textfield **obj**.

j_setfontname
procedure j_setfontname (integer obj , integer name)
Changes the font to the given **name**.

j_setfont
procedure j_setfont (integer obj , integer name , integer style , integer size)
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize
procedure j_setfontsize (integer obj , integer size)
Changes the font to the given **size**.

j_setfontstyle
procedure j_setfontstyle (integer obj , integer style)
Changes the font to the given **style**.

j_setnamedcolorbg
procedure j_setnamedcolorbg (integer obj , integer color)
Sets the background color to a predefined **color**.

j_setnamedcolor
procedure j_setnamedcolor (integer obj , integer color)
Sets the foreground color to a predefined **color**.

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

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

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

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

Vscrollbar

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds vscrollbar **obj** to container **cont**

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to vscrollbar **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables vscrollbar **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the vscrollbar **obj**.

j_enable *procedure j_enable (integer obj)*
enables the vscrollbar **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to vscrollbar **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of vscrollbar **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*
Returns the total pixel height of the actual font of vscrollbar **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of vscrollbar **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getvalue *integer function j_getvalue (integer obj)*
Returns the current setting of the scrollbar.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of vscrollbar **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of vscrollbar **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of vscrollbar **obj** in its parent's coordinate space.

j_hide *procedure j_hide (integer obj)*
Hides the vscrollbar **obj**.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to vscrollbar **obj**, and returns its event number.

j_mouselistener *integer function j_mouselistener (integer obj , integer kind)*
Adds a new mouse listener to vscrollbar **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_popupmenu *integer function j_popupmenu (integer obj , character*(*) label)*
Creates a new popupmenu with the specified **label** and returns its event number.

j_print *procedure j_print (integer obj)*
prints the vscrollbar .

j_release *procedure j_release (integer obj)*
Releases vscrollbar **obj** from its parent component (container).

j_setblockinc *integer function j_setblockinc (integer obj , integer val)*
Changes the block increment amount for the vscrollbar to **val**.

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves vscrollbar **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the vscrollbar 's **obj** cursor to the specified **cursor**.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to vscrollbar **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setmax *integer function j_setmax (integer obj , integer val)*
Changes the maximum value for the vscrollbar to **val**.

j_setmin *integer function j_setmin (integer obj , integer val)*
Changes the minimum value for the vscrollbar to **val**.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

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

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

j_setslidesize *integer function j_setslidesize (integer obj , integer val)*
Changes the slide size to **val**.

j_setunitinc *integer function j_setunitinc (integer obj , integer val)*
Changes the unit increment amount for the vscrollbar to **val**

j_setvalue *procedure j_setvalue (integer obj , integer val)*
Changes the current value of the vscrollbar to **val**.

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

Window

j_window *integer function j_window (integer obj)*
Creates a new simple window and returns its event number.

j_add *procedure j_add (integer obj , integer cont)*
Adds window **obj** to container **cont**

j_borderpanel *integer function j_borderpanel (integer obj , integer type)*
Creates a new borderpanel component with the style **type** and returns its event number.

j_button *integer function j_button (integer obj , character∗(∗) label)*
Creates a new button component with the specified **label** and returns its event number.

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

j_checkbox *integer function j_checkbox (integer obj , character∗(∗) label)*
Creates a new checkbox component with the specified **label** and returns its event number.

j_choice *integer function j_choice (integer obj)*
Creates a new choice component and returns its event number.

j_componentlistener *integer function j_componentlistener (integer obj , integer kind)*
Adds a new componentlistener to window **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_disable *procedure j_disable (integer obj)*
Disables window **obj** so that it is unresponsive to user interactions

j_dispose *procedure j_dispose (integer obj)*
Releases the resources of the window **obj**.

j_enable *procedure j_enable (integer obj)*
enables the window **obj**.

j_focuslistener *integer function j_focuslistener (integer obj)*
Adds a new focus listener to window **obj**, and returns its event number.

j_getfontascent *integer function j_getfontascent (integer obj)*
Returns the ascent (space above the baseline) of the actual font of window **obj**.

j_getfontheight *integer function j_getfontheight (integer obj)*

Returns the total pixel height of the actual font of window **obj**.

j_getheight *integer function j_getheight (integer obj)*
Returns the height of window **obj**.

j_getinsets *integer function j_getinsets (integer obj , integer side)*
Returns the width of the specified inset.

j_getlayoutid *integer function j_getlayoutid (integer obj)*
Returns the event number of the layoutmanager for containers **obj**.

j_getparentid *integer function j_getparentid (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getparent *integer function j_getparent (integer obj)*
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

j_getwidth *integer function j_getwidth (integer obj)*
Returns the width of window **obj**.

j_getxpos *integer function j_getxpos (integer obj)*
Returns the current horizontal position of window **obj** in its parent's coordinate space.

j_getypos *integer function j_getypos (integer obj)*
Returns the current vertical position of window **obj** in its parent's coordinate space.

j_graphicbutton *integer function j_graphicbutton (integer obj , character∗(∗) filename)*
Creates a new graphicbutton component with the image loaded from **filename** and returns its event number.

j_graphiclabel *integer function j_graphiclabel (integer obj , character∗(∗) str)*
Creates a new graphiclabel component with the image loaded from **filename** and returns its event number.

j_hide *procedure j_hide (integer obj)*
Hides the window **obj**.

j_hscrollbar *integer function j_hscrollbar (integer obj)*
Creates a new horizontal scrollbar and returns its event number.

j_isparent *integer function j_isparent (integer obj , integer cont)*
Returns .true. if **cont** is parent of **obj**, .false. otherwise.

j_isvisible *integer function j_isvisible (integer obj)*
Returns .true. if **obj** is visible, .false. otherwise.

j_keylistener *integer function j_keylistener (integer obj)*
Adds a new key listener to window **obj**, and returns its event number.

j_label
integer function j_label (integer obj , character∗(∗) label)
Creates a new label component with the specified **label** and returns its event number.

j_led
integer function j_led (integer obj , integer style , integer color)
Creates a new led component with the specified **style** and the specified color **color**.

j_line
integer function j_line (integer obj , integer orient , integer style , integer length)
Creates a new line component with the specified **length** and returns its event number.

j_list
integer function j_list (integer obj , integer rows)
Creates a new list component with the specified number of **rows** and returns its event number.

j_meter
integer function j_meter (integer obj , character∗(∗) title)
Creates a new pointer–intrument with the specified label **titel**.

j_mouselistener
integer function j_mouselistener (integer obj , integer kind)
Adds a new mouse listener to window **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

j_pack
procedure j_pack (integer obj)
Resizes window to the minimal size of contained components.

j_panel
integer function j_panel (integer obj)
Creates a new panel component and returns its event number.

j_popupmenu
integer function j_popupmenu (integer obj , character∗(∗) label)
Creates a new popupmenu with the specified **label** and returns its event number.

j_print
procedure j_print (integer obj)
prints the window .

j_progressbar
integer function j_progressbar (integer obj , integer orient)
Creates a new progressbar with the specified **orient**ation.

j_radiogroup
integer function j_radiogroup (integer obj)
Creates a new radiogroup and returns its event number.

j_releaseall
procedure j_releaseall (integer obj)
Releases all components from window **obj**.

j_release
procedure j_release (integer obj)
Releases window **obj** from its parent component (container).

j_scrollpane
integer function j_scrollpane (integer obj)
Creates a new scrollpane component and returns its event number.

j_setalign
procedure j_setalign (integer obj , integer align)

Sets the alignment in window **obj** to **align**. Needs a flowlayout Manager.

j_setborderlayout *procedure j_setborderlayout (integer obj)*
Adds a borderlayout manager to window **obj**.

j_setborderpos *procedure j_setborderpos (integer obj , integer pos)*
Moves window **obj** at a certain position. The outer container needs a border layout manager.

j_setcolorbg *procedure j_setcolorbg (integer obj , integer r , integer g, , integer b)*
Sets the background color to the (**r, g, b**) values.

j_setcolor *procedure j_setcolor (integer obj , integer r , integer g, , integer b)*
Sets the foreground color to the (**r, g, b**) values.

j_setcursor *integer function j_setcursor (integer obj , integer cursor)*
Changes the window 's **obj** cursor to the specified **cursor**.

j_setfixlayout *procedure j_setfixlayout (integer obj)*
Adds a fixlayout manager to window **obj** (default layout manager).

j_setflowfill *procedure j_setflowfill (integer obj , integer bool)*
Resizes all containing component to the height (width) of window **obj**. Needs a flowlayout manager.

j_setflowlayout *procedure j_setflowlayout (integer obj , integer align)*
Adds a flowlayout manager to window **obj** with the specified **align**ment.

j_setfocus *integer function j_setfocus (integer obj)*
Directs the input focus to window **obj**.

j_setfontname *procedure j_setfontname (integer obj , integer name)*
Changes the font to the given **name**.

j_setfont *procedure j_setfont (integer obj , integer name , integer style , integer size)*
Changes the font to the given characteristics **name**, **style** and **size**.

j_setfontsize *procedure j_setfontsize (integer obj , integer size)*
Changes the font to the given **size**.

j_setfontstyle *procedure j_setfontstyle (integer obj , integer style)*
Changes the font to the given **style**.

j_setgridlayout *procedure j_setgridlayout (integer obj , integer row , integer col)*
Adds a gridlayout manager to window **obj** with the specified **row**s and **col**umns.

j_sethgap *procedure j_sethgap (integer obj , integer hgap)*
Sets the horizontal gap between components to **hgap** Pixel.

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

Set the insets to the specified values.

j_setnamedcolorbg *procedure j_setnamedcolorbg (integer obj , integer color)*
Sets the background color to a predefined **color**.

j_setnamedcolor *procedure j_setnamedcolor (integer obj , integer color)*
Sets the foreground color to a predefined **color**.

j_setnolayout *procedure j_setnolayout (integer obj)*
Removes the current layout manager from window **obj** .

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

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

j_setvgap *procedure j_setvgap (integer obj , integer vgap)*
Sets the vertical gap between components to **hgap** Pixel.

j_sevensegment *integer function j_sevensegment (integer obj , integer color)*
Creates a new sevensegment display with the specified color **color**.

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

j_textarea *integer function j_textarea (integer obj , integer rows , integer columns)*
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

j_textfield *integer function j_textfield (integer obj , integer columns)*
Creates a new textfield component with the specified number of **columns** and returns its event number.

j_vscrollbar *integer function j_vscrollbar (integer obj)*
Creates a new vertical scrollbar and returns its event number.

j_windowlistener *integer function j_windowlistener (integer window , integer kind)*
Adds a new windowlistener to **obj**, and returns its event number. An event occures, if the user action is of kind **kind**.

Kapitel 2

Functions

additem

Synopsis

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

Arguments

obj	integer
str	character∗(∗)

Description

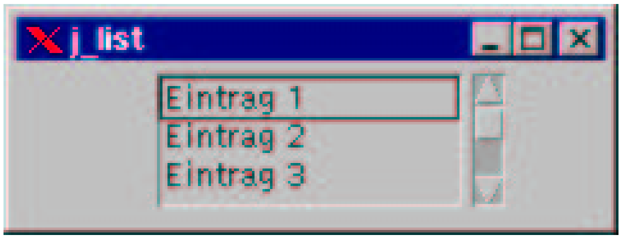
adds a new item containing **str** to component **obj**.

Targets

List, Choice

Example

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



add

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

Arguments

obj	integer
cont	integer

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

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

alertbox

Synopsis

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

Arguments

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

Description

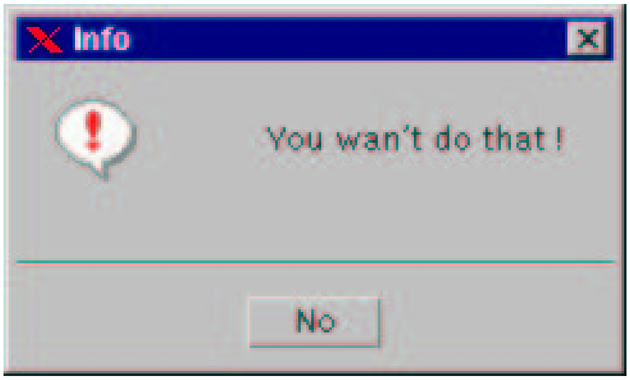
Shows a alertbox with the specified **title**, **text** and **button**. Alertboxes are modal dialogs, the application is blocked until the button or the closeicon is clicked. The return value is 0 if the closeicon is clicked and 1 if the buttons is used.

Targets

Frame

Example

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



You wan't do that !

appendtext

Synopsis	procedure **j_appendtext** (integer obj , character∗(∗) text)
Arguments	obj integer text character∗(∗)
Description	Appends the given **text** to the **obj** current text.
Targets	Textarea

beep

Synopsis procedure **j_beep** ( )

Description Emits an audio beep.

borderpanel

Synopsis

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

Arguments

obj	integer
type	integer

Description

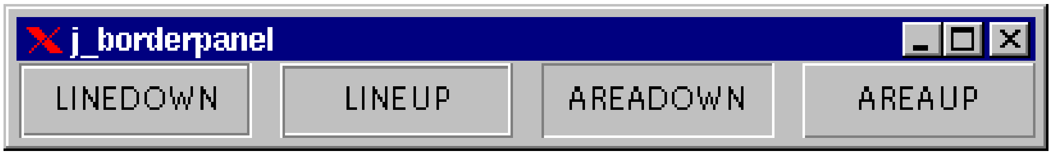
Creates a new borderpanel component with the style **type** and returns its event number.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



button

Synopsis integer function **j_button** (integer obj , character∗(∗) label)

Arguments

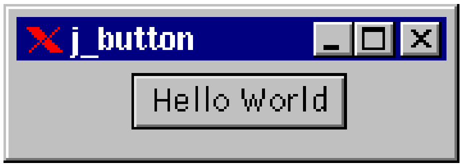
obj	integer
label	character∗(∗)

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



canvas

Synopsis

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

Arguments

obj	integer
width	integer
height	integer

Description

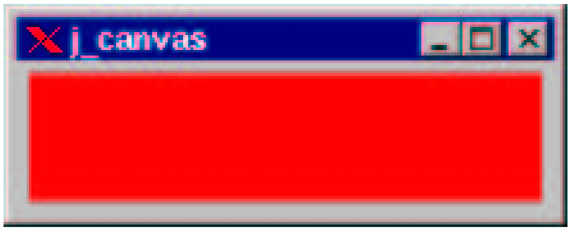
Creates a new canvas component with the given **width** and **height** and returns its event number. A canvas can be used for general drawing functions. A canvas generates an event, if its size changes. On error −1 will be returned.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



checkbox

Synopsis

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

Arguments

obj	integer
label	character∗(∗)

Description

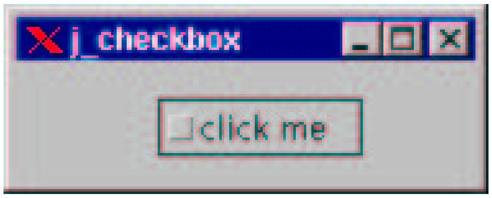
Creates a new checkbox component with the specified **label** and returns its event number.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



checkmenuitem

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

Arguments

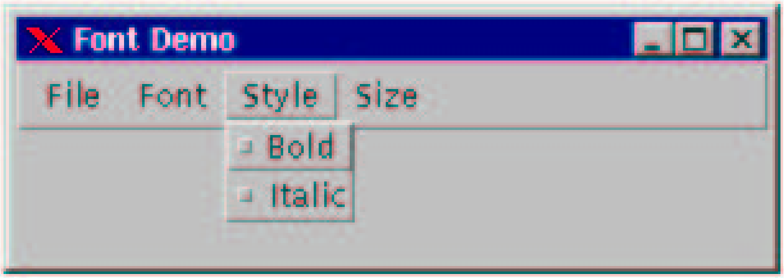
obj	integer
label	character*(*)

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

Targets Menu, Popupmenu, Helpmenu

Example

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



choicebox2

Synopsis

procedure **j_choicebox2** (integer obj , character∗(∗) title , character∗(∗) text , character∗(∗) button1 , character∗(∗) button2)

Arguments

obj	integer
title	character∗(∗)
text	character∗(∗)
button1	character∗(∗)
button2	character∗(∗)

Description

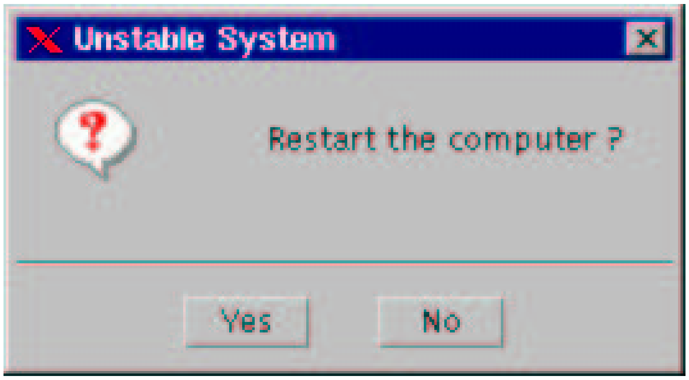
Shows a choicebox with the specified **title**, **text** and two buttons. Choiceboxes are modal dialogs, the application is blocked until a button or the closeicon is clicked. The focus is set to the first button. The return value is 0 if the closeicon is clicked, 1 for the first button and 2 for the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

Targets

Frame

Example

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



Restart the computer ?

choicebox3

Synopsis

procedure **j_choicebox3** (integer obj , character*(*) title , character*(*) text , character*(*) button1 , character*(*) button2 , character*(*) button3)

Arguments

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

Description

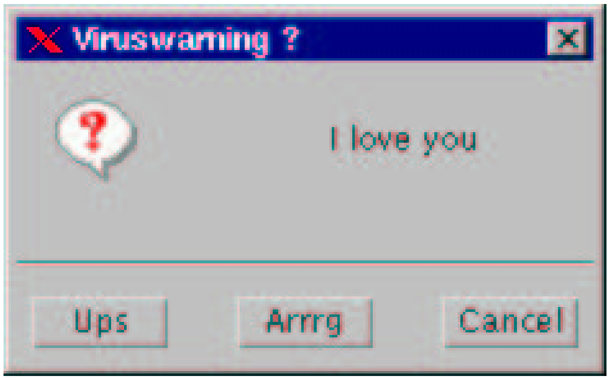
Shows a choicebox with the specified **title**, **text** and three buttons. Choiceboxes are modal dialogs, the application is blocked until a button or the closeicon is clicked. The focus is set to the first button. The return value is 0 if the closeicon is clicked, 1 for the first button, 2 for the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

Targets

Frame

Example

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



I love you

choice

Synopsis integer function **j_choice** (integer obj)

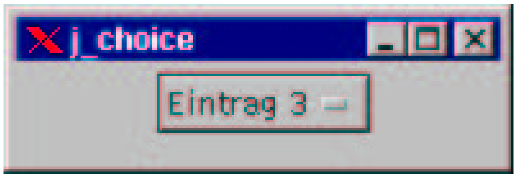
Arguments obj integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



cliprect

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

componentlistener

Synopsis

integer function **j_componentlistener** (integer obj , integer kind)

Arguments

obj	integer
kind	integer

Description

Adds a new componentlistener to component **obj**, and returns its event number. An event occures, if the user action is of kind **kind**. Posible values for **kind**:

- J_RESIZED : An event occures when the component has been resized.
- J_HIDDEN : An event occures when the component has been hidden.
- J_SHOWN : An event occures when the component has been shown.

Targets

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

connect

Synopsis integer function **j_connect** (character∗(∗) hostname)

Arguments hostname character∗(∗)

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

Example

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

    or

if( .not. j_connect("127.0.0.1")) then
:
```

delete

Synopsis

procedure **j_delete** (integer obj , integer start , integer end)

Arguments

obj	integer
start	integer
end	integer

Description

Delets text from starting position **start** to ending position **end**.

Targets

Textarea

deselect

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

dialog

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

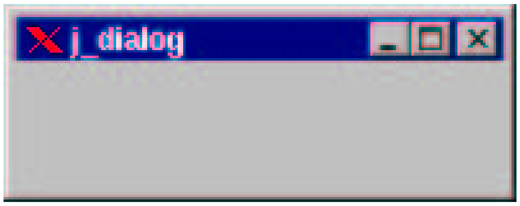
Arguments		
	obj	integer
	label	character*(*)

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

Targets Frame

Example

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



disable

Synopsis	procedure **j_disable** (integer obj)
Arguments	obj integer
Description	Disables component **obj** so that it is unresponsive to user interactions
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menuitem, CheckBoxMenuitem,Menu, HelpMenu, Popupmenu

dispose

Synopsis	procedure **j_dispose** (integer obj)	
Arguments	obj	integer
Description	Releases the resources of the component **obj**.	
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Canvas, Image, Printer, Keylistener, Focuslistener, Mouselistener	

drawarc

Synopsis

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

Arguments

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

Description

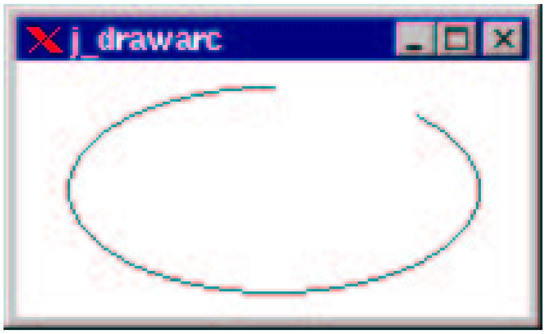
Draws an unfilled arc from angle **arc1** to angle **arc2** with the center (**x, y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets

Canvas, Image, Printer

Example

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



drawcircle

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
r	integer

Description

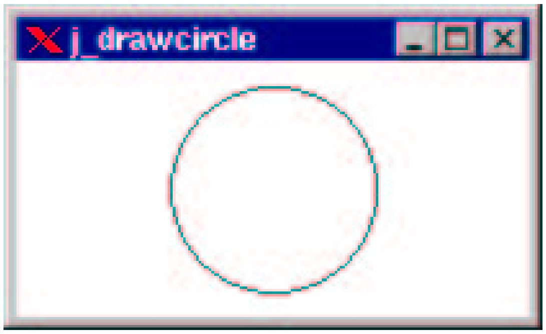
Draws an unfilled circle with center (**x, y**) and radius **x**.

Targets

Canvas, Image, Printer

Example

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



drawimagesource

Synopsis

procedure **j_drawimagesource** (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)

Arguments

obj	integer
x	integer
y	integer
w	integer
h	integer
r	array of integer
g	array of integer
b	array of integer

Description

Paints an image at Position (**x, y,**) with **w**idth and **h**eight. The red, green and blue values of each pixel are given by the arrays **r, g, b**.

Targets

Canvas, Image, Printer

drawimage

Synopsis

procedure **j_drawimage** (integer obj , integer image , integer x , integer y)

Arguments

obj	integer
image	integer
x	integer
y	integer

Description

Copies the image, given by its eventnumber **image**, to position (**x, y**).

Targets

Canvas, Image, Printer

drawline

Synopsis

procedure **j_drawline** (integer obj , integer x1 , integer y1 , integer x2 , integer y2)

Arguments

obj	integer
x1	integer
y1	integer
x2	integer
y2	integer

Description

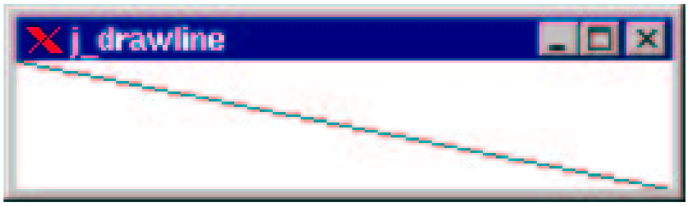
Draws a line connecting **(x1,y1)** and **(x2,y2)**.

Targets

Canvas, Image, Printer

Example

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



drawoval

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
rx	integer
ry	integer

Description

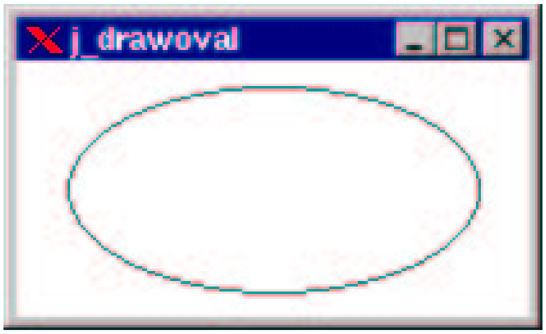
Draws an unfilled oval with the center (**x, y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets

Canvas, Image, Printer

Example

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



drawpixel

Synopsis

procedure **j_drawpixel** (integer obj , integer x , integer y)

Arguments

obj	integer
x	integer
y	integer

Description

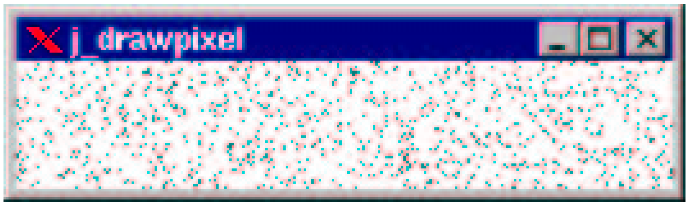
Draws a pixel at **(x,y)**.

Targets

Canvas, Image, Printer

Example

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



drawpolygon

Synopsis

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

Arguments

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

Description

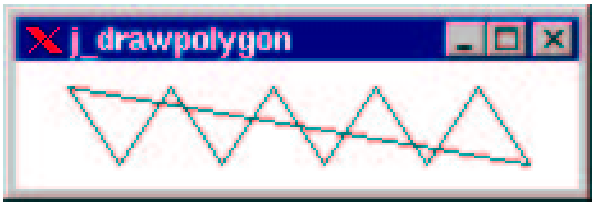
Draws an unfilled polygon based on first **len** elements in **x** and **y**.

Targets

Canvas, Image, Printer

Example

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



drawpolyline

Synopsis

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

Arguments

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

Description

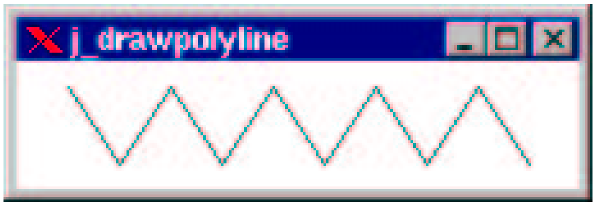
Draws a series of line segments based on first **len** elements in **x** and **y**.

Targets

Canvas, Image, Printer

Example

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



drawrect

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
width	integer
height	integer

Description

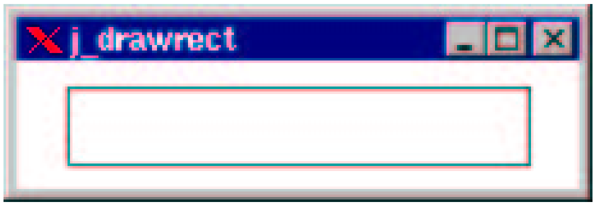
Draws an unfilled rectangle from **(x,y)** of size **width** x **height**.

Targets

Canvas, Image, Printer

Example

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



drawroundrect

Synopsis

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

Arguments

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

Description

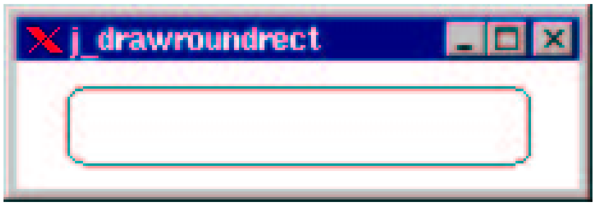
Draws an unfilled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

Targets

Canvas, Image, Printer

Example

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



drawscaleddimage

Synopsis

procedure **j_drawscaleddimage** (integer obj , integer image , integer sx , integer sy , integer sw , integer sh , integer tx , integer ty , integer tw , integer th)

Arguments

obj	integer
image	integer
sx	integer
sy	integer
sw	integer
sh	integer
tx	integer
ty	integer
tw	integer
th	integer

Description

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

Targets

Canvas, Image, Printer

drawstring

Synopsis

procedure **j_drawstring** (integer obj , integer x , integer y , character∗(∗) str)

Arguments

obj	integer
x	integer
y	integer
str	character∗(∗)

Description

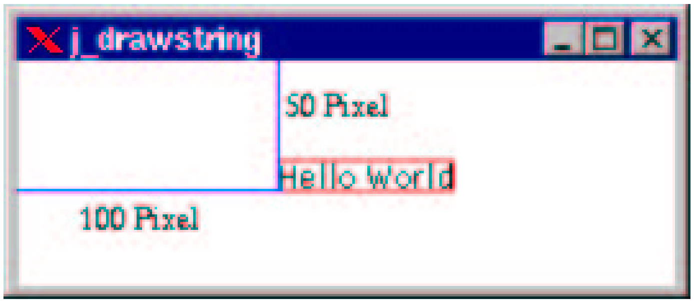
Draws text on screen at position (**x,y**).

Targets

Canvas, Image, Printer

Example

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



enable

Synopsis	procedure **j_enable** (integer obj)
Arguments	obj integer
Description	enables the component **obj**.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menuitem, CheckBoxMenuitem,Menu, HelpMenu, Popupmenu

filedialog

Synopsis

procedure **j_filedialog** (integer frame , character∗(∗) title , character∗(∗) directory , character∗(∗) filename)

Arguments

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

Description

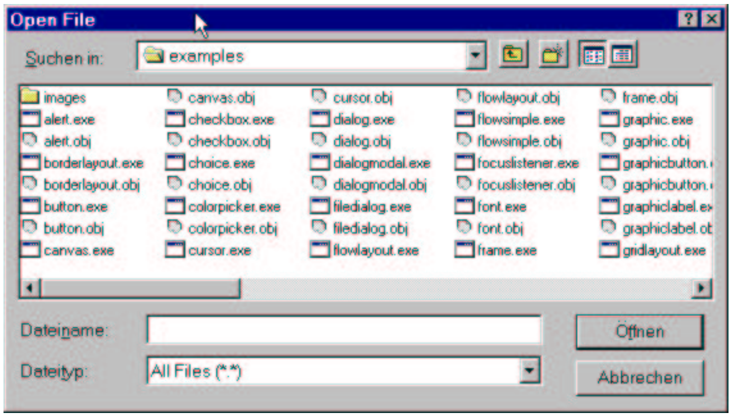
Opens a filedialog box in the specified **directory** with the specified **title** and returns the selected **filename**. If **title** contains "/S" the SAVE–filedialog will be called. The substring "/S" will be removed.

Targets

Frame

Example

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



fileselector

Synopsis

procedure **j_fileselector** (integer frame , character∗(∗) title , character∗(∗) filter , character∗(∗) filename)

Arguments

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

Description

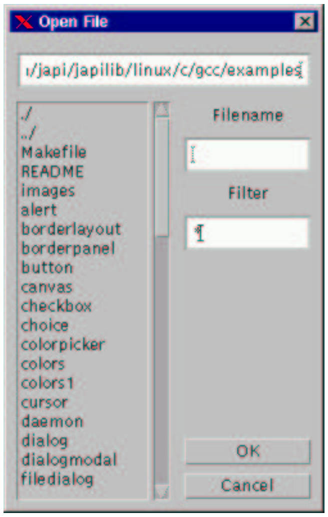
Opens a fileslector box with the preselected **filename** and the specified **title** and returns the selected **filename**. **filter** specifies the Filename Filter. A Fileselector can be used with output redirections via j_connect();

Targets

Frame

Example

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



ı/japi/japilib/linux/c/gcc/examples

./
../
Makefile
README
images
alert
borderlayout
borderpanel
button
canvas
checkbox
choice
colorpicker
colors
colors1
cursor
daemon
dialog
dialogmodal
filedialog

fillarc

Synopsis

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

Arguments

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

Description

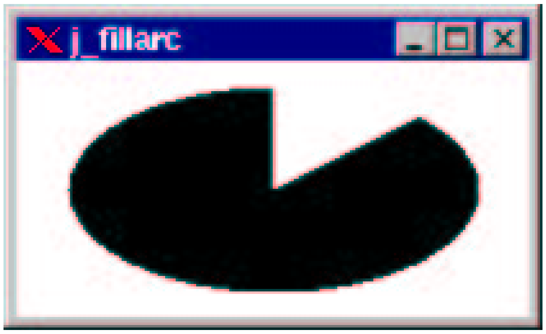
Draws an filled arc from angle **arc1** to angle **arc2** with the center (**x, y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets

Canvas, Image, Printer

Example

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



fillcircle

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
r	integer

Description

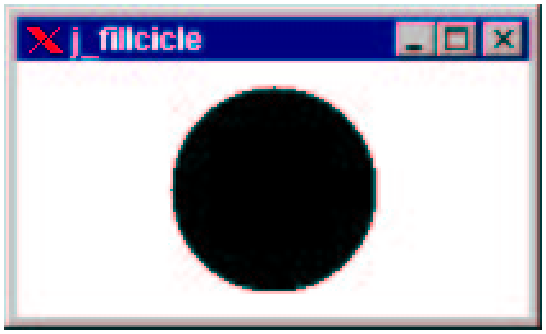
Draws an filled circle with center (**x, y**) and radius **x**.

Targets

Canvas, Image, Printer

Example

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



filloval

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
rx	integer
ry	integer

Description

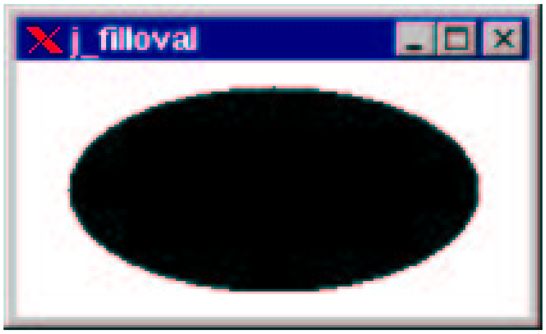
Draws an filled oval with the center (**x, y**) and the horizontal radius **rx** and the vertical radius **ry**.

Targets

Canvas, Image, Printer

Example

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



fillpolygon

Synopsis

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

Arguments

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

Description

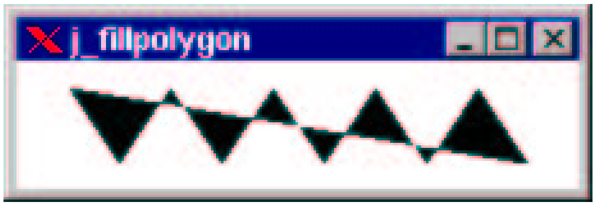
Draws an filled polygon based on first **len** elements in **x** and **y**.

Targets

Canvas, Image, Printer

Example

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



fillrect

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
width	integer
height	integer

Description

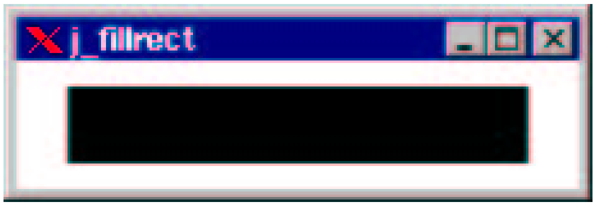
Draws an filled rectangle from **(x,y)** of size **width** x **height**.

Targets

Canvas, Image, Printer

Example

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



fillroundrect

Synopsis

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

Arguments

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

Description

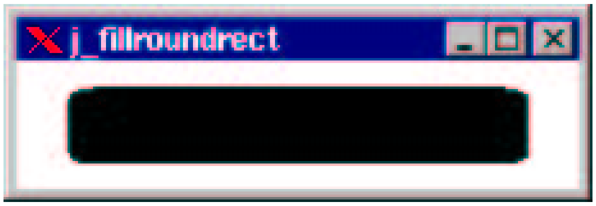
Draws an filled rectangle from **(x,y)** of size **width** x **height** with rounded corners. **arcx** and **arcy** specify the radius of rectangle corners.

Targets

Canvas, Image, Printer

Example

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



focuslistener

Synopsis integer function **j_focuslistener** (integer obj)

Arguments obj integer

Description Adds a new focus listener to component **obj**, and returns its event number.

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

frame

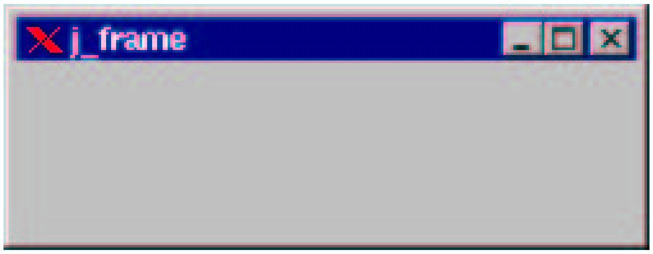
Synopsis integer function **j_frame** (character*(*) label)

Arguments label character*(*)

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

Example

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



getaction

Synopsis integer function **j_getaction** ( )

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

getcolumns

Synopsis procedure **j_getcolumns** (integer obj)

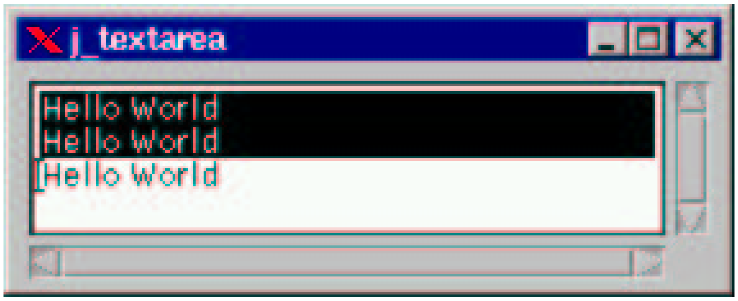
Arguments obj integer

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

Targets Textarea, Textfield, Gridlayout

Example

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



Hello World
Hello World
Hello World

getcurpos

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

getfontascent

Synopsis	integer function **j_getfontascent** (integer obj)	
Arguments	obj	integer

Description

Returns the ascent (space above the baseline) of the actual font of component **obj**.

Targets

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

getfontheight

Synopsis integer function **j_getfontheight** (integer obj)

Arguments obj integer

Description Returns the total pixel height of the actual font of component **obj**.

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

getheight

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

Arguments obj integer

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

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

Example

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

> 22
```

getimagesource

Synopsis

integer function **j_getimagesource** (integer obj , integer x , integer y , integer w , integer h , array of integer r , array of integer g , array of integer b)

Arguments

obj	integer
x	integer
y	integer
w	integer
h	integer
r	array of integer
g	array of integer
b	array of integer

Description

Returns an image of the specified size (**x, y, w**idth, **h**eight) of component . The red, green and blue values of each pixel will be stored in **r, g, b**

Targets

Canvas, Image

getimage

Synopsis integer function **j_getimage** (integer obj)

Arguments obj integer

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

Targets Canvas, Image

getinsets

Synopsis

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

Arguments

obj	integer
side	integer

Description

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

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

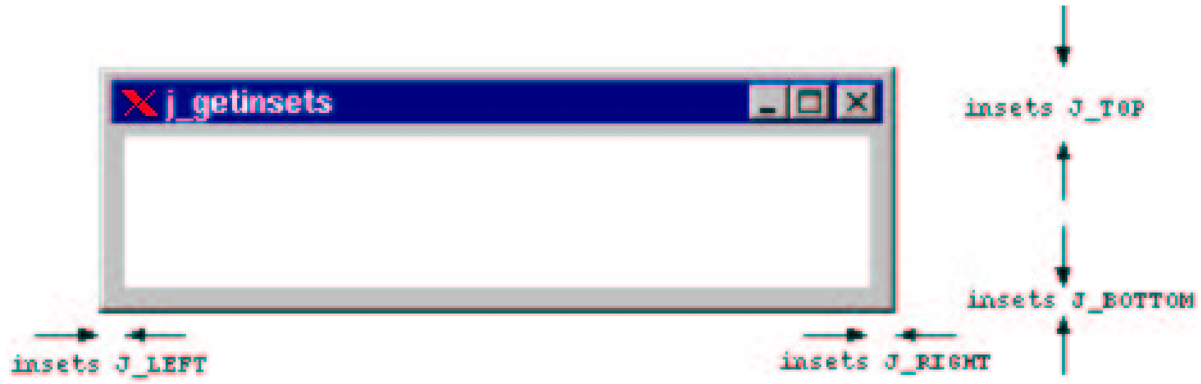
Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
frame  = j_frame("j_getinsets")
write (*,*) j_getinsets(frame,J_TOP),j_getinsets(frame,J_BOTTOM),
            j_getinsets(frame,J_LEFT),j_getinsets(frame,J_RIGHT)
:

> 25 5 5 6
```



getitemcount

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

Arguments obj integer

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

Targets List, Choice

getitem

Synopsis procedure **j_getitem** (integer obj , integer item , character∗(∗) str)

Arguments

	obj	integer
	item	integer
	str	character∗(∗)

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

Targets List, Choice

getkeychar

Synopsis	integer function **j_getkeychar** (integer obj)
Arguments	obj integer
Description	Returns the ascii value of the last pressed key.
Targets	Keylistener

getkeycode

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

getlayoutid

Synopsis

integer function **j_getlayoutid** (integer obj)

Arguments

obj	integer

Description

Returns the event number of the layoutmanager for containers **obj**.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
call j_setgridlayout(frame,2,2)
grid = j_getlayoutid(frame)
:
```

getlength

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

getmousebutton

Synopsis integer function **j_getmousebutton** (integer mouselistener)

Arguments mouselistenerinteger

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

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

Targets Mouselistener

getmousex

Synopsis integer function **j_getmousex** (integer mouselistener)

Arguments mouselistenerinteger

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

Targets Mouselistener

getmousey

Synopsis integer function **j_getmousey** (integer mouselistener)

Arguments mouselistenerinteger

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

Targets Mouselistener

getparentid

Synopsis integer function **j_getparentid** (integer obj)

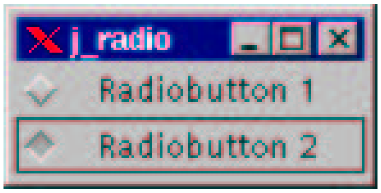
Arguments obj integer

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

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, Menuitem, CheckBox-Menuitem,Menu, HelpMenu, Popupmenu, Radiogroup

Example

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



getparent

Synopsis

integer function **j_getparent** (integer obj)

Arguments

obj	integer

Description

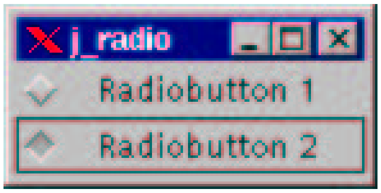
Returns the parent event number of component **obj**. If **obj** is a frame −1 will be returned.

Targets

Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, Menuitem, CheckBox-Menuitem,Menu, HelpMenu, Popupmenu, Radiogroup

Example

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



getrows

Synopsis procedure **j_getrows** (integer obj)

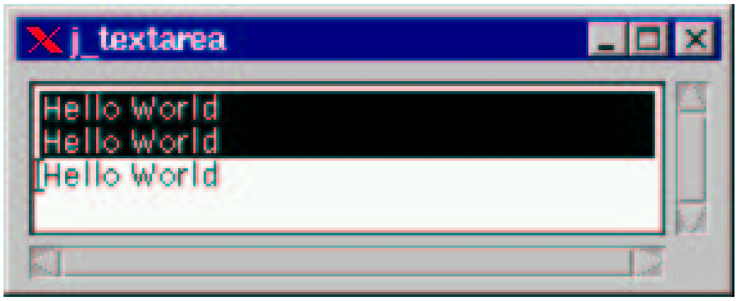
Arguments obj integer

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

Targets Textarea, Gridlayout

Example

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



Hello World
Hello World
Hello World

getscaledimage

Synopsis

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

Arguments

obj	integer
x	integer
y	integer
sw	integer
sh	integer
tw	integer
th	integer

Description

Copy the contents of the rectangular area defined by **x, y,** width **sw**, and height **sh** into an image and return its eventnumber. The image will be scaled to target width **th** and target height **th**.

Targets

Canvas, Image

getscreenheight

Synopsis integer function **j_getscreenheight** ( )

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

Example

```
:
write(*,*) j_getscreenwidth(), j_getscreenheight()
:

> 1280 1024
```

getscreenwidth

Synopsis

integer function **j_getscreenwidth** ( )

Description

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

Example

```
:
write(*,*) j_getscreenwidth(), j_getscreenheight()
:

> 1280 1024
```

getselect

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

getselend

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

getselstart

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

getseltext

Synopsis	procedure **j_getseltext** (integer obj , character*(*) text)
Arguments	obj integer text character*(*)
Description	Returns the currently selected text of component **obj**.
Targets	Textarea, Textfield

getstate

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

gettext

Synopsis procedure **j_gettext** (integer obj , character∗(∗) str)

Arguments

obj	integer
str	character∗(∗)

Description returns the component 's text or label.

Targets Button, Label, Checkbox, Radiobutton, Dialog, Frame, Menuitem, CheckBoxMenuitem,Menu, HelpMenu, Popupmenu, Textarea, Textfield

Example

```
character*256 str
:
label = j_label(frame,"Hello World")
call j_gettext(label,str)
write(*,*) str
:

> Hello World
```

getvalue

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

Arguments obj integer

Description Returns the current setting of the scrollbar.

Targets Scrollbar

getviewportheight

Synopsis integer function **j_getviewportheight** (integer obj)

Arguments obj integer

Description Returns the height of the component 's **obj** port (the area that is shown)

Targets Scrollpane

getviewportwidth

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

getwidth

Synopsis integer function **j_getwidth** (integer obj)

Arguments obj integer

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

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

Example

```
:
label = j_getlabel(frame,"Hello World")
write(*,*) j_getwidth(label))
:

> 84
```

getxpos

Synopsis integer function **j_getxpos** (integer obj)

Arguments obj integer

Description Returns the current horizontal position of component **obj** in its parent's coordinate space.

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

getypos

Synopsis	integer function **j_getypos** (integer obj)	
Arguments	obj	integer

Description

Returns the current vertical position of component **obj** in its parent's coordinate space.

Targets

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

graphicbutton

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

Arguments

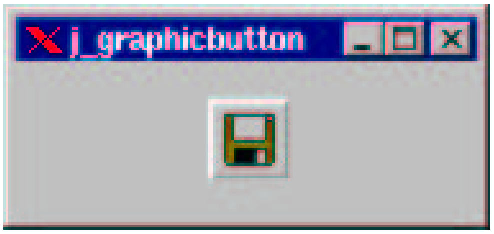
obj	integer
filename	character*(*)

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



graphiclabel

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

Arguments

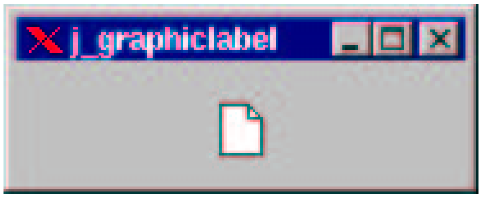
obj	integer
str	character*(*)

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



hasfocus

Synopsis integer function **j_hasfocus** (integer obj)

Arguments obj integer

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

Targets Focuslistener

helpmenu

Synopsis

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

Arguments

obj	integer
label	character∗(∗)

Description

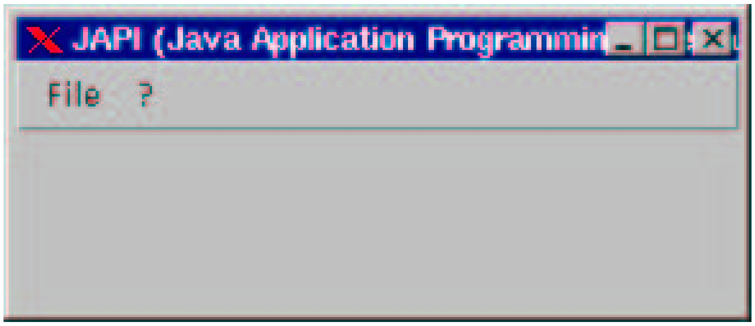
Creates a new helpmenu component with the specified **label** and returns its event number.

Targets

Menubar

Example

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



hide

Synopsis	procedure **j_hide** (integer obj)
Arguments	obj integer
Description	Hides the component **obj**.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

hscrollbar

Synopsis integer function **j_hscrollbar** (integer obj)

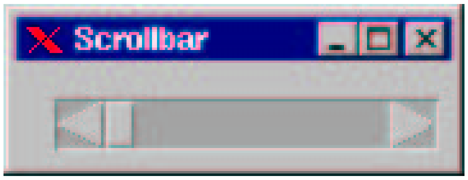
Arguments obj integer

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

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

Example

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



image

Synopsis

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

Arguments

width	integer
height	integer

Description

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

Example

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

insert

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

inserttext

Synopsis procedure **j_inserttext** (integer obj , character*(*) text , integer pos)

Arguments

	obj	integer
	text	character*(*)
	pos	integer

Description Places additional text within the component at the given position **pos**.

Targets Textarea

isparent

Synopsis integer function **j_isparent** (integer obj , integer cont)

Arguments

obj	integer
cont	integer

Description Returns .true. if **cont** is parent of **obj**, .false. otherwise.

Targets Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menubar, Menuitem, CheckBox-Menuitem,Menu, HelpMenu, Popupmenu, Radiogroup

isresizable

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

isselect

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

Arguments

obj	integer
item	integer

Description Returns .true. if the particular **item** is currently selected, .false. otherwise.

Targets List

isvisible

Synopsis	integer function **j_isvisible** (integer obj)
Arguments	obj integer
Description	Returns .true. if **obj** is visible, .false. otherwise.
Targets	Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment

keylistener

Synopsis integer function **j_keylistener** (integer obj)

Arguments obj integer

Description Adds a new key listener to component **obj**, and returns its event number.

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

label

Synopsis

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

Arguments

obj	integer
label	character*(*)

Description

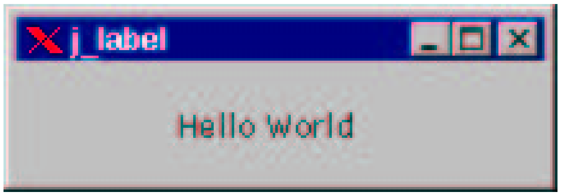
Creates a new label component with the specified **label** and returns its event number.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



led

Synopsis

integer function **j_led** (integer obj , integer style , integer color)

Arguments

obj	integer
style	integer
color	integer

Description

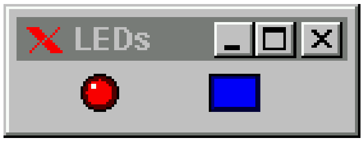
Creates a new led component and returns its event number. The LEDs shape could be round if **style**=J_ROUND or a rectangle if **style**=J_RECT. The color could be one of the predefined colors (eg. J_RED, J_GREEN).

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



line

Synopsis

integer function **j_line** (integer obj , integer orient , integer style , integer length)

Arguments

obj	integer
orient	integer
style	integer
length	integer

Description

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

- J_HORIZONTAL : horizontal line
- J_VERTICAL : vertical line

The Parameter **style** specifies the linestyle:

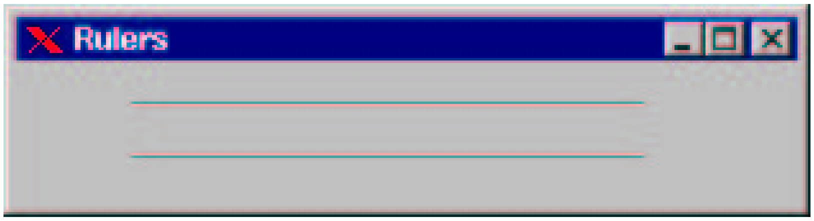
- J_LINEDOWN : etched-in linestyle.
- J_LINEUP : etchet-out linestyle.

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

integer function **j_list** (integer obj , integer rows)

Arguments

obj	integer
rows	integer

Description

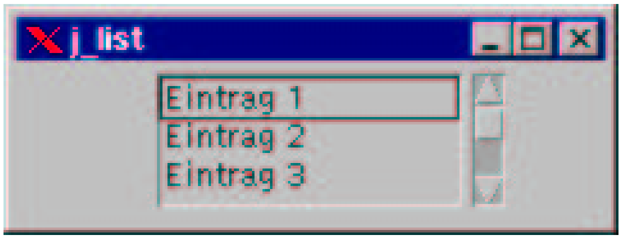
Creates a new list component with the specified number of **rows** and returns its event number.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



loadimage

Synopsis

integer function **j_loadimage** (character*(*) filename)

Arguments

filename character*(*)

Description

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

- GIF
- JPEG
- BMP
- PPM

Example

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

menubar

Synopsis integer function **j_menubar** (integer obj)

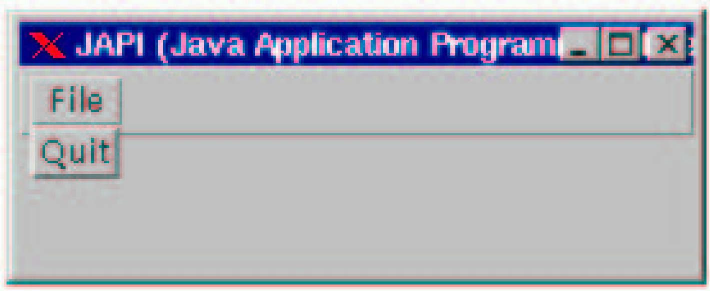
Arguments obj integer

Description Creates a new menubar and returns its event number.

Targets Frame

Example

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



menuitem

Synopsis

integer function **j_menuitem** (integer obj , character∗(∗) label)

Arguments

obj	integer
label	character∗(∗)

Description

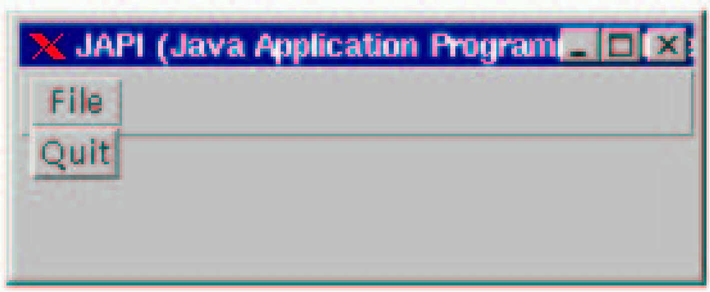
Creates a new menuitem with the specified **label** and returns its event number.

Targets

Menu, Popupmenu, Helpmenu

Example

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



menu

Synopsis integer function **j_menu** (integer obj , character∗(∗) str)

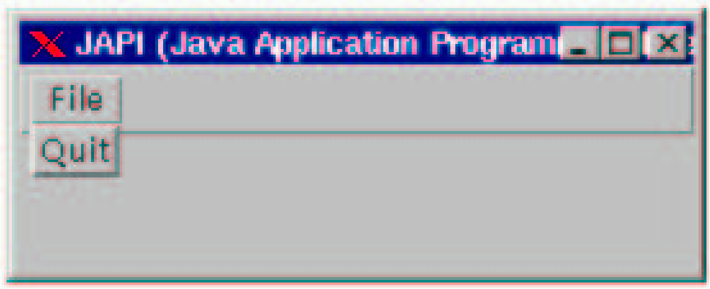
Arguments		
	obj	integer
	str	character∗(∗)

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

Targets Menubar, Menu

Example

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



messagebox

Synopsis

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

Arguments

obj	integer
title	character∗(∗)
text	character∗(∗)

Description

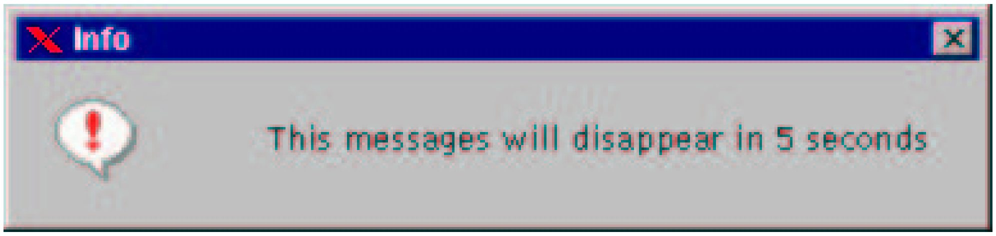
Shows a messagebox with the specified **title** and **text** and returns its event number. In the case of error −1 will be returend. A Messagebox generates an event, if the close icon is clicked.

Targets

Frame

Example

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



This messages will disappear in 5 seconds

meter

Synopsis

integer function **j_meter** (integer obj , character*(*) title)

Arguments

obj	integer
title	character*(*)

Description

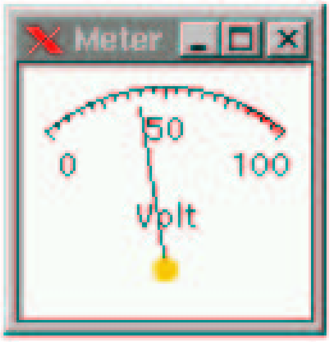
Creates a new pointer–intrument with the specified label **titel** and returns its event number. The meter has predifiend values from 0 to 100. This can be canged via $j_setmin()$ and $j_setmax()$. A danger value is set to 80 and can be justified with $j_setdanger()$

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
meter = j_meter(frame,"Volt")
call j_setvalue(meter,90)
:
```



mouselistener

Synopsis

integer function **j_mouselistener** (integer obj , integer kind)

Arguments

obj	integer
kind	integer

Description

Adds a new mouse listener to component **obj**, and returns its event number. An event occures, if the user action is of kind **kind**. Posible values for **kind**:

- J_ENTERED : An event occures if the mouse cursor has been moved into the component **obj**.
- J_MOVED : An event occures if the mouse cursor has been moved inside the component **obj**.
- J_EXITED : An event occures if the mouse cursor has been moved out of the component **obj**.
- J_PRESSED : An event occures if a mouse button was pressed.
- J_DRAGGED : An event occures if the mouse cursor has been dragged (moved with pressed button) inside the component **obj**.
- J_RELEASED : An event occures if a mouse button was released.
- J_DOUBLECLICK : An event occures if a mouse button was doubleclicked.

Targets

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

multiplemode

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

Arguments	obj	integer
	bool	integer

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

Targets List

nextaction

Synopsis integer function **j_nextaction** ( )

Description Waits for the next event.

pack

Synopsis procedure **j_pack** (integer obj)

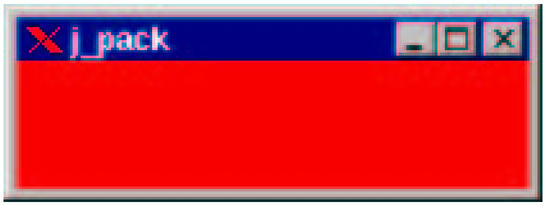
Arguments obj integer

Description Resizes component to the minimal size of contained components.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



panel

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

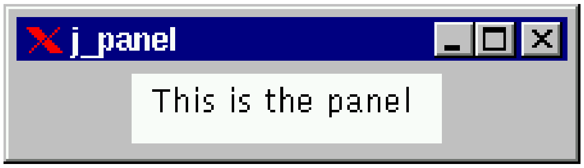
Arguments obj integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



This is the panel

popupmenu

Synopsis

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

Arguments

obj	integer
label	character∗(∗)

Description

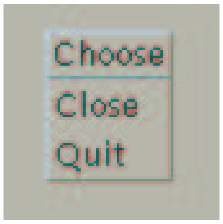
Creates a new popupmenu with the specified **label** and returns its event number.

Targets

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

Example

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



printer

Synopsis

integer function **j_printer** (integer frame)

Arguments

frame	integer

Description

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

Targets

Frame

Example

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

print

Synopsis procedure **j_print** (integer obj)

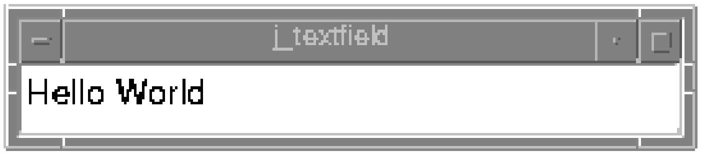
Arguments obj integer

Description prints the component .

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

Example

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



Hello World

progressbar

Synopsis

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

Arguments

obj	integer
orient	integer

Description

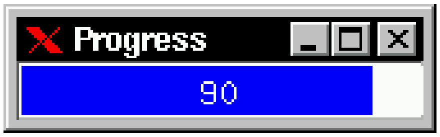
Creates a new progressbar with the specified **orient**ation and returns its event number. Orientation could be J_HORIZONTAL or J_VERTICAL. The progressbar has predifiend values from 0 to 100. This can be canged via *j_setmin*() and *j_setmax*().

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



quit

Synopsis procedure **j_quit** ( )

Description Cancels the connection to the JAPI Kernel.

radiobutton

Synopsis

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

Arguments

obj	integer
label	character*(*)

Description

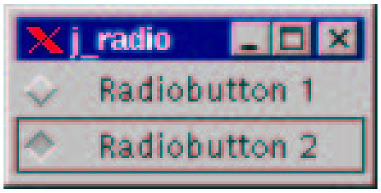
Creates a new radiobutton with the specified **label** and returns its event number.

Targets

Radiogroup

Example

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



radiogroup

Synopsis integer function **j_radiogroup** (integer obj)

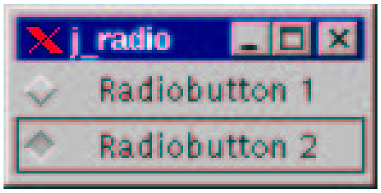
Arguments obj integer

Description Creates a new radiogroup and returns its event number.

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



random

Synopsis integer function **j_random** ( )

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

releaseall

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

release

Synopsis procedure **j_release** (integer obj)

Arguments obj integer

Description Releases component **obj** from its parent component (container).

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

removeall

Synopsis	integer function **j_removeall** (integer obj)
Arguments	obj integer
Description	Removes all items from the component .
Targets	List, Choice

removeitem

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

remove

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

Arguments	obj	integer
	item	integer

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

Targets List, Choice

replacetext

Synopsis

procedure **j_replacetext** (integer obj , character*(*) text , integer start , integer end)

Arguments

obj	integer
text	character*(*)
start	integer
end	integer

Description

Replaces the text from starting position **start** to ending position **end** with the given **text**.

Targets

Textarea

saveimage

Synopsis

integer function **j_saveimage** (integer obj , character*(*) filename , integer filetyp)

Arguments

obj	integer
filename	character*(*)
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) .eq. false)
    write (*,*) "Error saving Bitmap file"
:
```

scrollpane

Synopsis integer function **j_scrollpane** (integer obj)

Arguments

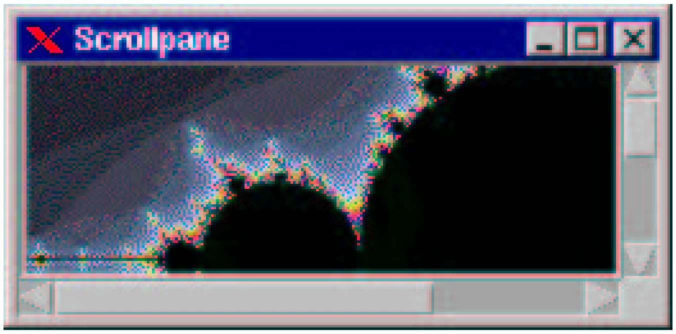
obj	integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



selectall

Synopsis	procedure **j_selectall** (integer obj)
Arguments	obj integer
Description	Selects all the text in the component .
Targets	Textarea, Textfield

select

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

Arguments	obj	integer
	item	integer

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

Targets List, Choice

selecttext

Synopsis

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

Arguments

obj	integer
start	integer
end	integer

Description

Selects text from starting position **start** to ending position **end**.

Targets

Textarea, Textfield

seperator

Synopsis procedure **j_seperator** (integer obj)

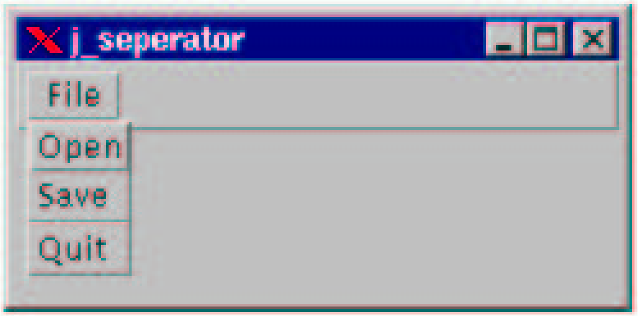
Arguments obj integer

Description Adds a separator bar to the component .

Targets Menu, HelpMenu, Popupmenu

Example

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



setalign

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

setblockinc

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

Arguments		
	obj	integer
	val	integer

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

Targets Scrollbar

setborderlayout

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

setborderpos

Synopsis procedure **j_setborderpos** (integer obj , integer pos)

Arguments

obj	integer
pos	integer

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

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

setcolorbg

Synopsis

procedure **j_setcolorbg** (integer obj , integer r , integer g, , integer b)

Arguments

obj	integer
r	integer
g,	integer
b	integer

Description

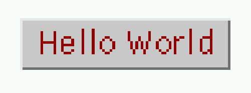
Sets the background color to the (**r, g, b**) values.

Targets

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

Example

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



setcolor

Synopsis

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

Arguments

obj	integer
r	integer
g,	integer
b	integer

Description

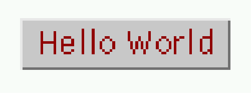
Sets the foreground color to the (**r, g, b**) values.

Targets

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

Example

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



setcolumns

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

Arguments

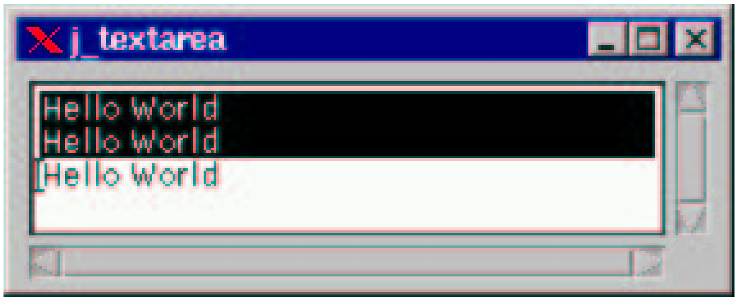
obj	integer
columns	integer

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

Targets Textarea, Textfield, Gridlayout

Example

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



setcurpos

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

setcursor

Synopsis	integer function **j_setcursor** (integer obj , integer cursor)	
Arguments	obj	integer
	cursor	integer

Description

Changes the component 's **obj** cursor to the specified **cursor**.

Targets

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

setdebug

Synopsis	procedure **j_setdebug** (integer level)
Arguments	level integer
Description	Sets the debuglevel to **level**.

setechochar

Synopsis procedure **j_setechochar** (integer obj , character chr)

Arguments

obj	integer
chr	character

Description Changes the character **chr** that is used to echo all user input in the component .

Targets Textfield

seteditable

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

Arguments

obj	integer
bool	integer

Description Allows to make the component editable (**bool**=.true.) or read-only (**bool**=.false.).

Targets Textarea, Textfield

setfixlayout

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

setflowfill

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

setflowlayout

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

setfocus

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

setfontname

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

Arguments

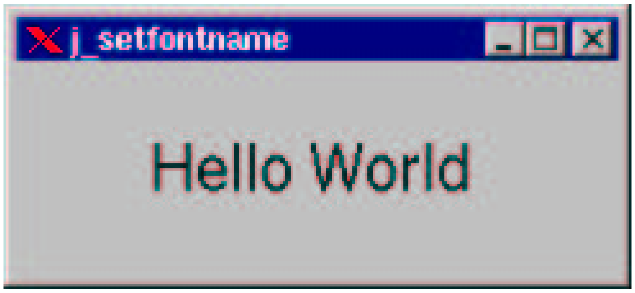
obj	integer
name	integer

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

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

Example

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



setfontsize

Synopsis procedure **j_setfontsize** (integer obj , integer size)

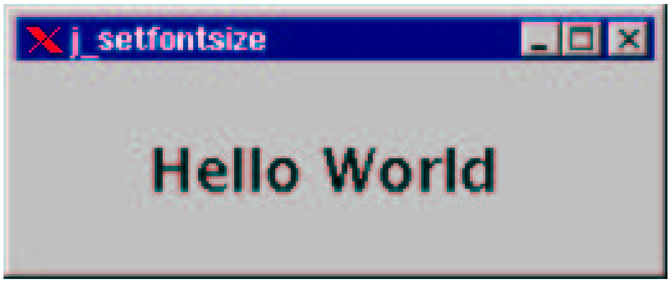
Arguments		
	obj	integer
	size	integer

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

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

Example

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



setfontstyle

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

Arguments

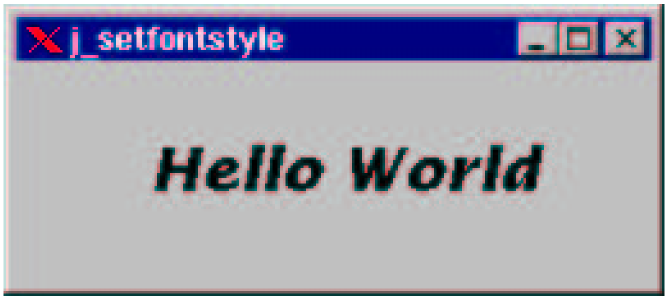
obj	integer
style	integer

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

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

Example

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



setfont

Synopsis

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

Arguments

obj	integer
name	integer
style	integer
size	integer

Description

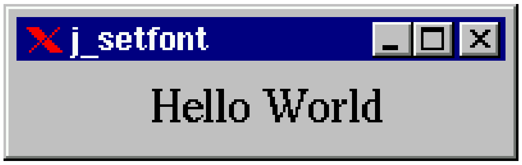
Changes the font to the given characteristics **name**, **style** and **size**.

Targets

Button, Graphicbutton, Canvas, Checkbox, Radiobutton, Choice, Label, Graphiclabel, List, Scrollbar, Panel, Borderpanel, Window, Dialog, Frame, Scrollpane, Textarea, Textfield, Led, Progressbar, Meter, Sevensegment, Menuitem, CheckBoxMenuitem,Menu, HelpMenu, Popupmenu

Example

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



Hello World

setgridlayout

Synopsis

procedure **j_setgridlayout** (integer obj , integer row , integer col)

Arguments

obj	integer
row	integer
col	integer

Description

Adds a gridlayout manager to component **obj** with the specified **row**s and **col**umns.

Targets

Panel, Borderpanel, Window, Dialog, Frame

sethgap

Synopsis

procedure **j_sethgap** (integer obj , integer hgap)

Arguments

obj	integer
hgap	integer

Description

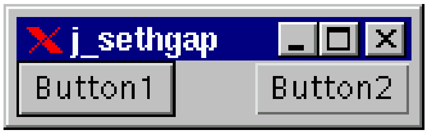
Sets the horizontal gap between components to **hgap** Pixel.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



seticon

Synopsis

procedure **j_seticon** (integer frame , integer icon)

Arguments

frame	integer
icon	integer

Description

Sets the image **icon** to display when the **frame** is iconized. Not all platforms support the concept of iconizing a window.

Targets

Frame

Example

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

setimage

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

Arguments

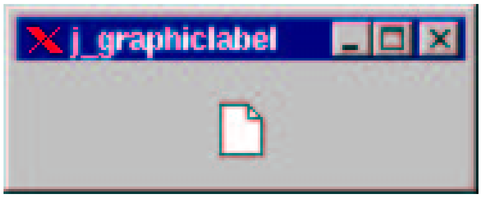
obj	integer
image	integer

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

Targets Graphicbutton, Graphiclabel

Example

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



setinsets

Synopsis

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

Arguments

obj	integer
top	integer
bottom	integer
left	integer
right	integer

Description

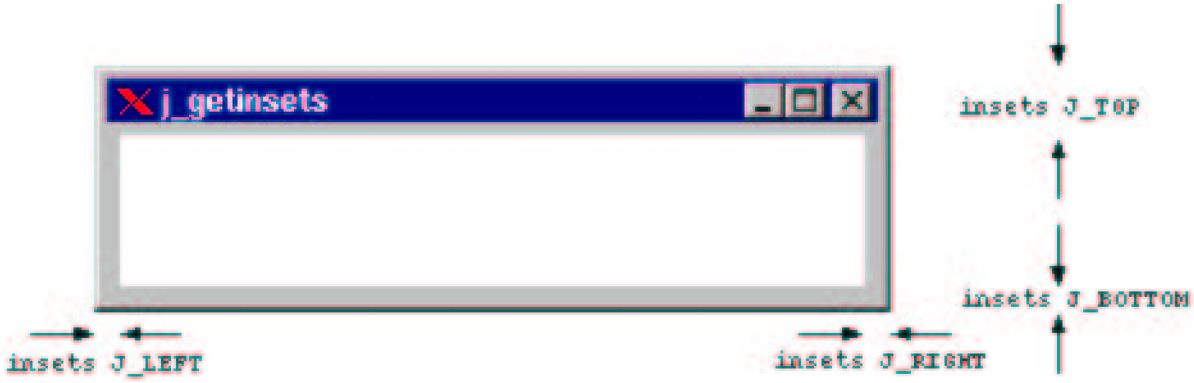
Set the insets to the specified values.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

```
:
frame  = j_frame("j_getinsets")
call j_setinsets(frame,30,10,10,10)
:
> 25 5 5 6
```



setmax

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

Arguments

obj	integer
val	integer

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

Targets Scrollbar, Meter, Progress

setmin

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

setnamedcolorbg

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

setnamedcolor

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

setnolayout

Synopsis procedure **j_setnolayout** (integer obj)

Arguments obj integer

Description Removes the current layout manager from component **obj** .

Targets Panel, Borderpanel, Window, Dialog, Frame

setpos

Synopsis

procedure **j_setpos** (integer obj , integer xpos , integer ypos)

Arguments

obj	integer
xpos	integer
ypos	integer

Description

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

Targets

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

setradiogroup

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

setresizable

Synopsis	procedure **j_setresizable** (integer obj , integer resizable)
Arguments	obj integer resizable integer
Description	The component cannot be resized, if **resizable** is .false. .
Targets	Dialog, Frame
Example	

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

setrows

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

Arguments

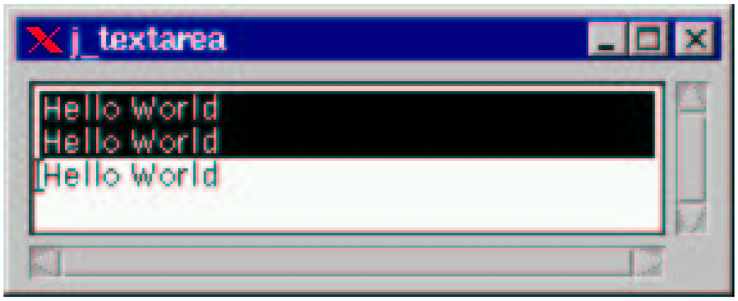
obj	integer
rows	integer

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

Targets Textarea, Gridlayout

Example

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



Hello World
Hello World
Hello World

setshortcut

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

setsize

Synopsis

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

Arguments

obj	integer
width	integer
height	integer

Description

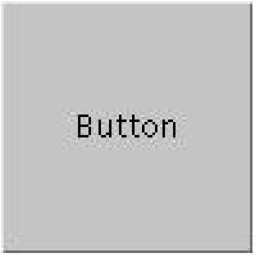
Resizes component **obj** to specified **width** and **height**.

Targets

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

Example

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



setslidesize

Synopsis	integer function **j_setslidesize** (integer obj , integer val)	
Arguments	obj	integer
	val	integer
Description	Changes the slide size to **val**.	
Targets	Scrollbar	

setstate

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

Arguments

obj	integer
bool	integer

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

Targets Checkbox, Radiobutton, Checkmenuitem, Led

settext

Synopsis

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

Arguments

obj	integer
str	character*(*)

Description

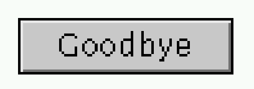
Sets the content or the label of the component **obj** to **str**.

Targets

Button, Label, Checkbox, Radiobutton, Dialog, Frame, Menuitem, CheckBoxMenuitem,Menu, HelpMenu, Popupmenu, Textarea, Textfield

Example

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



setunitinc

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

Arguments

obj	integer
val	integer

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

Targets Scrollbar

setvalue

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

setvgap

Synopsis procedure **j_setvgap** (integer obj , integer vgap)

Arguments

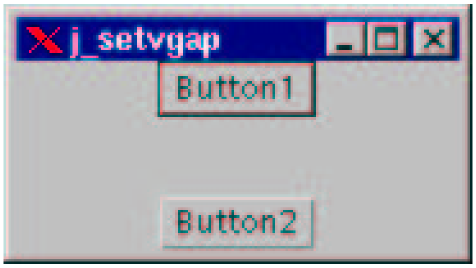
obj	integer
vgap	integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



setxor

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

Arguments		
	obj	integer
	bool	integer

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

Targets Canvas, Image, Printer

sevensegment

Synopsis

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

Arguments

obj	integer
color	integer

Description

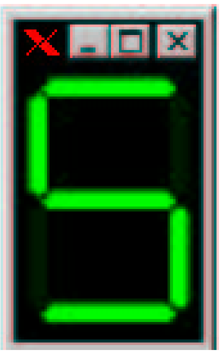
Creates a new sevensegment display and returns its event number. The color could be one of the predefined colors (eg. J_RED, J_GREEN).

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



showpopup

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

show

Synopsis procedure **j_show** (integer obj)

Arguments obj integer

Description Shows the component **obj**.

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

sleep

Synopsis integer function **j_sleep** (integer msec)

Arguments msec integer

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

start

Synopsis integer function **j_start** ( )

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

Example

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

sync

Synopsis procedure **j_sync** ( )

Description Synchronizes the application with the JAPI kernel.

textarea

Synopsis

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

Arguments

obj	integer
rows	integer
columns	integer

Description

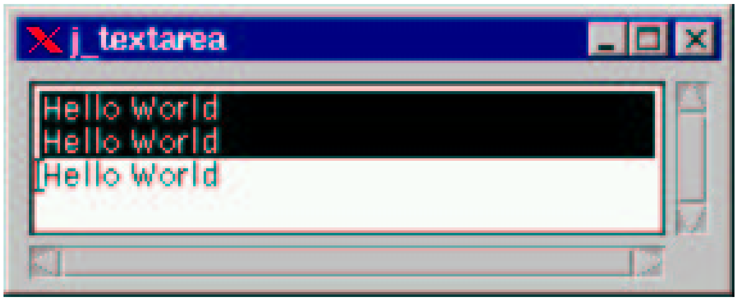
Creates a new textarea component with the specified number of **rows columns** and returns its event number.

Targets

Panel, Borderpanel, Window, Dialog, Frame

Example

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



```
Hello World
Hello World
Hello World
```

textfield

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

Arguments

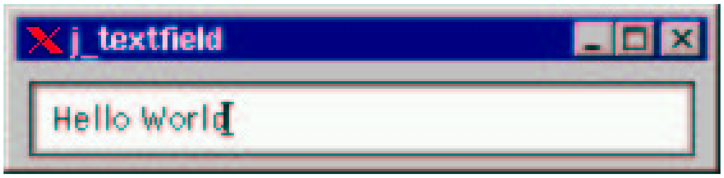
obj	integer
columns	integer

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

Targets Panel, Borderpanel, Window, Dialog, Frame

Example

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



Hello World

translate

Synopsis procedure **j_translate** (integer obj , integer x , integer y)

Arguments		
	obj	integer
	x	integer
	y	integer

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

Targets Canvas, Image, Printer

vscrollbar

Synopsis integer function **j_vscrollbar** (integer obj)

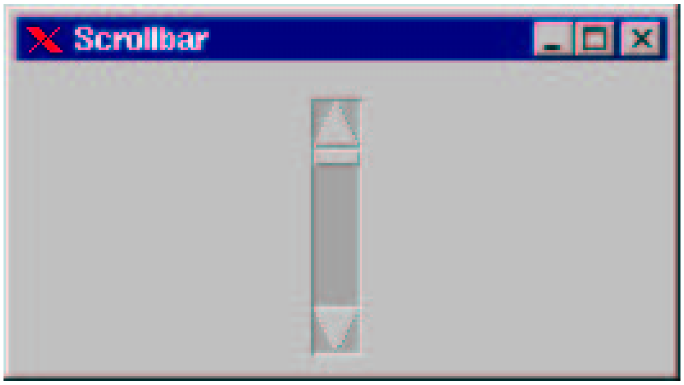
Arguments obj integer

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

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

Example

```
:
scroll=j_vscrollbar(frame)
call j_setpos(scroll,120,40)
call j_setsize(scroll,20,100)
:
```



windowlistener

Synopsis

integer function **j_windowlistener** (integer window , integer kind)

Arguments

window	integer
kind	integer

Description

Adds a new windowlistener to component **obj**, and returns its event number. An event occures, if the user action is of kind **kind**. Posible values for **kind**:

- J_ACTIVATED : An event occures when the component is activated.
- J_DEACTIVATED : An event occures when the component is deactivated.
- J_OPENED : An event occures when the component has been opened.
- J_CLOSED : An event occures when the component has been closed.
- J_ICONFIED : An event occures when the component is iconfied.
- J_DEICONFIED : An event occures when the component is deiconfied.
- J_CLOSING : An event occures when the close icon has been clicked .

Targets

Window, Dialog, Frame

window

Synopsis integer function **j_window** (integer obj)

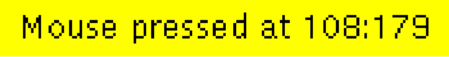
Arguments obj integer

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

Targets Frame

Example

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



Mouse pressed at 108:179