SIZE

Applies to

All controls, and system control.

Description

Returns or sets the size and position of the specified control.

Usage

objectsize = **Get_Property** (**"SIZE"**)

existingprop = **Set_Property** (*objectname*, **"SIZE"**, *newsize*)

Remarks

Values passed in Set_Property:

Value	Description
newsize	x:@FM:y:@FM:w:@FM:h:@fm:v

Returns

Values returned by both Get_Property and Set_Property:

Value	Description
objectsize	x:@FM:y:@FM:w:@FM:h[:@fm:v]
existingprop	Size and position, when Set_Property was run.

Set_Property values are:

Value	Description
X	Offset from the left side of the window.
Y	Offset from the top of the window.
W	Width (rightward extension) of the control.
H	Height (downward extension) of the control, where the top of the control is the y position.
V	Visible. If set to -1 and the VISIBLE property of the window is false, the window will remain invisible.

Get_Property returns x:@FM:y:@FM:w:@FM:h, where:

Value	Description
X	Offset from the left side of the window.
Y	Offset from the top of the window.
W	Width (rightward extension) of the control.
H	Height (downward extension) of the control, where the top of the control is the y position.

The SIZE property for SYSTEM returns:

Value	Description
<1>	Screen width.
<2>	Screen height.

<3>	Maximum window client area width.
<4>	Maximum window client area height.

Set_Property sets the size and position of the specified control. Set any of the four arguments to equal 32,768, if you want to ensure that the dimension does not change.

Note: If the control was hidden, applying the SIZE property makes it visible, except when setting the 5th field to -1.

See also

CLIENTSIZE property, Send_Message COLWIDTH, MAXIMIZESIZE property, TRACKINGSIZE property

Example

```
subroutine PlaceDialog(xPos, yPos)

* this subroutine places a dialog on the screen using the
* same information that Popup uses to determine its
* placement;
* xPos and yPos are passed in as:
*   -2     - center of screen
*   -1     - center of window
*   other  - offset to window
*
* assumes @window is the dialog id, which is true if this is
* called from an event, like the CREATE event, of a dialog

declare function   Get_Property
declare subroutine Set_Property

ScreenSize = Get_Property("SYSTEM", "SIZE")
DialogSize = Get_Property(@window, "SIZE")
Parent     = Get_Property(@window, "PARENT")
ParentSize = Get_Property(Parent, "SIZE")

wScreen    = ScreenSize<3>   ;* width of screen
hScreen    = ScreenSize<4>   ;* height of screen
xDialog    = DialogSize<1>   ;* position of dialog
yDialog    = DialogSize<2>   ;* position of dialog
wDialog    = DialogSize<3>   ;* width of dialog
hDialog    = DialogSize<4>   ;* height of dialog
xParent    = ParentSize<1>   ;* position of parent
yParent    = ParentSize<2>   ;* position of parent
wParent    = ParentSize<3>   ;* width of parent
hParent    = ParentSize<4>   ;* height of parent

* calculate position for the dialog begin case
  * center w.r.t. screen
  case xPos = -2 or yPos = -2
    xDialog = (wScreen - wDialog) / 2
    yDialog = (hScreen - hDialog) / 2
  * center w.r.t. parent
  case xPos = -1 or yPos = -1
    xDialog = xParent + (wParent - wDialog) / 2
    yDialog = yParent + (hParent - hDialog) / 2
  * position w.r.t. parent
  case 1
    xDialog = xParent + xPos
    yDialog = yParent + yPos
end case

* make sure that the dialog is on the screen
if xDialog < 0                then xDialog = 0
if yDialog < 0                then yDialog = 0
if xDialog + wDialog > wScreen then xDialog = wScreen -
Dialog
if yDialog + hDialog > hScreen then yDialog = hScreen - hDialog
  NewSize = xDialog: @fm: yDialog: @fm: wDialog: @fm: hDialog
  Set_Property(@window, "SIZE", NewSize)
return
```