

Rectangles

Rectangles serve as decoration and can be displayed in a frame or in a frame background. Rectangles are display-only, and you cannot tab to a rectangle. Rectangles can receive user input only if they are movable or resizable. See [Chapter 24, “Direct Manipulation,”](#) for more information on moving and resizing widgets. Also, as previously mentioned, you can define ToolTip values for rectangles.

In a rectangle, the foreground color (FGCOLOR) is used for the edge and the background color (BGCOLOR) is used to fill the interior. In a character interface, the display color (DCOLOR) is used to fill the interior. The default values for unspecified colors are the owning frame’s foreground and background colors. You can use the EDGE-PIXELS or EDGE-CHARS attributes to set the thickness of the rectangle’s edge. The default edge is 1 pixel in graphical interfaces and 1 character in character interfaces. Use the NO-FILL option to create a hollow, transparent rectangle.

NOTE: The minimum size for a rectangle in character mode is 1 character high and 1 character wide. If you specify the size in pixels, Progress rounds down to a lower character. If the result is a dimension of less than 1 character, Progress does not display the rectangle.

If you specify an edge width of 0 and you do not specify a background color, you will not see the rectangle because it inherits the background color from its frame. To see the rectangle, you must either set the background color or specify an edge width that is greater than 0.

[Table 18–1](#) shows the various effects that you can achieve by using different combinations of the FGCOLOR, BGCOLOR, and NO-FILL options in a graphical interface. The table assumes that the application’s color table contains red in the first location and blue in the second.

Table 18–1: Using NO-FILL and Color with Rectangles

Options	Resulting Rectangle
No options	A rectangle filled with the frame’s background color and bordered in the frame’s foreground color.
FGCOLOR 0	A rectangle with a red border filled with the frame’s background color.
NO-FILL	A hollow rectangle bordered with the frame’s foreground color.
FGCOLOR 0 NO-FILL	A hollow rectangle with a red border.
FGCOLOR 0 BGCOLOR 1	A rectangle with a red border filled with blue.
BGCOLOR 1	A rectangle filled with blue and bordered with the frame’s foreground color.

For example, the following code fragment defines a hollow rectangle with a border that is either one pixel (in graphical interfaces) or one character (in character interfaces) thick. The border is drawn in the frame’s foreground color. If you want a solid fill rectangle without a border, specify 0 for the EDGE-PIXELS or EDGE-CHARS attribute.

NOTE: In a character interface, a rectangle’s proportions are determined by the proportions of the character set. For example, if a character set’s height is two times the width, specify a SIZE option of 2 BY 1 if you want a square.

Some character interfaces support line graphics characters. If you want Progress to use these characters when drawing a rectangle, specify the GRAPHIC-EDGE option in the DEFINE RECTANGLE statement or set the GRAPHIC-EDGE attribute to TRUE. The GRAPHIC-EDGE attribute is ignored for a minimum-sized character rectangle (1 character by 1 character). In a graphical interface, Progress ignores the GRAPHIC-EDGE option and attribute setting.

At run time, an application can query or set the rectangle’s fill or edge width using the FILLED and EDGE-PIXELS or EDGE-CHARS attributes.

Usually, you should put any rectangles into the frame background. If you put a rectangle in the foreground in character mode, you should display the rectangle before displaying anything else in the frame. This prevents the rectangle from overlaying other widgets.

DEFINE RECTANGLE Statement

Defines a rectangle widget for use in the current procedure.

SYNTAX

```
DEFINE RECTANGLE rectangle [ LIKE rectangle2 ]  
[ NO-FILL ]
```

[{ EDGE-CHARS *width* } | { EDGE-PIXELS *width* }]
[DCOLOR *expression*]
[BGCOLOR *expression*]
[FGCOLOR *expression*]
[GRAPHIC-EDGE]
[PFCOLOR *expression*]
[*size-phrase*]
[TOOLTIP *tooltip*]
{ [*trigger-phrase*] }

rectangle

The name of the rectangle you are defining.

LIKE *rectangle2*

Specifies a previously defined rectangle whose characteristics you want to apply to the new rectangle. If you name a rectangle with this option, you must have defined that rectangle previously in the procedure.

NO-FILL

Indicates that only the outline of the rectangle should be drawn. By default, the rectangle is filled with the background color.

EDGE-CHARS *width*

Specifies the width of the rectangle outline in characters. The default width is 1. If you do not want an edge on the rectangle, specify EDGE-CHARS 0.

EDGE-PIXELS *width*

Specifies the width of the rectangle outline in pixels. The default width is 1. If you do not want an edge on the rectangle, specify EDGE-PIXELS 0.

DCOLOR *expression*

Specifies the fill color of the rectangle in character interfaces. This option is ignored in graphical interfaces.

BGCOLOR *expression*

Specifies the background color or fill color of the rectangle in graphical interfaces. This option is ignored in character interfaces.

FGCOLOR *expression*

Specifies the foreground color or edge color of the rectangle in graphical interfaces. This option is ignored in character interfaces.

GRAPHIC-EDGE

Specifies that in a character interface, the rectangle is drawn with graphic characters. This option is ignored in a graphical interface. This overrides the EDGE-CHARS and

EDGE-PIXELS options. The border is one graphic unit thick.

PFCOLOR *expression*

Specifies the edge color of the rectangle in character interfaces. This option is ignored in graphical interfaces. It is also ignored if you specify GRAPHIC-EDGE.

size-phrase

Specifies the outside dimensions of the rectangle widget. This is the syntax for size-phrase. If you specify SIZE or SIZE-CHARS, the units are characters; if you specify SIZE-PIXELS, the units are pixels. For character units, the values *width* and *height* must be decimal constants. For pixels units, they must be integer constants. For more information, see the [SIZE Phrase](#) reference entry.

SYNTAX { SIZE | SIZE-CHARS | SIZE-PIXELS } *width* BY *height*

TOOLTIP *tooltip*

Allows you to define a help text message for a rectangle widget. Progress automatically displays this text when the user pauses the mouse button over the rectangle widget.

You can add or change the TOOLTIP option at any time. If TOOLTIP is set to "" or ? (the unknown value), then the ToolTip is removed. No ToolTip is the default. The TOOLTIP option is supported in Windows only.

trigger-phrase

Specifies application triggers for the rectangle.

For more information, see the [Trigger Phrase](#) reference entry.

CREATE Widget Statement

Creates a dynamic object, such as a widget object.

SYNTAX

```
CREATE { BUTTON | COMBO-BOX
| CONTROL-FRAME | DIALOG-BOX
| EDITOR | FILL-IN
| FRAME | IMAGE
| MENU | MENU-ITEM
| RADIO-SET | RECTANGLE
| SELECTION-LIST | SLIDER
| SUB-MENU | TEXT
| TOGGLE-BOX | WINDOW
| VALUE ( string-expression )
} widget-handle [ IN WIDGET-POOL pool-name ]
[ ASSIGN { attribute = expression } ... ]
[ trigger-phrase ]
```

VALUE (*string-expression*)

An expression of type CHARACTER that evaluates to the type of object you want to create (for example, BUTTON) with any combination of uppercase and lowercase characters.

widget-handle

A variable of type WIDGET-HANDLE that Progress sets to the value of the new widget handle.

IN WIDGET-POOL *pool-name*

Specifies the widget pool in which the object is created. If you do not specify a widget pool, the object is created in the current default widget pool.

ASSIGN { *attribute* = *expression* } ...

Assigns specified values to attributes of the object. The *attribute* parameter must be the name of a valid attribute for the object and *expression* must evaluate to a valid value for that attribute.

trigger-phrase

A trigger phrase associated with the object. For more information, see the [Trigger Phrase](#) reference entry.

RECTANGLE Widget

A rectangle is a graphical widget that can be displayed in a frame foreground or background. You can define a static rectangle with the DEFINE RECTANGLE statement. You can create a dynamic rectangle with the CREATE Widget statement.

Attributes

[BGCOLOR Attribute 1](#)

[COLUMN Attribute](#)

[DATA-TYPE Attribute](#)

[DCOLOR Attribute 2](#)

[DYNAMIC Attribute 2](#)

EDGE-CHARS Attribute
EDGE-PIXELS Attribute
FGCOLOR Attribute1
FILLED Attribute
FRAME Attribute
FRAME-COL Attribute 2
FRAME-NAME Attribute 2
FRAME-ROW Attribute 2
FRAME-X Attribute 2
FRAME-Y Attribute 2
GRAPHIC-EDGE Attribute 3
HANDLE Attribute2
HEIGHT-CHARS Attribute
HEIGHT-PIXELS Attribute
HELP Attribute
HIDDEN Attribute
HWND Attribute 2, 4
MANUAL-HIGHLIGHT Attribute 1
MOVABLE Attribute 1
NAME Attribute
NEXT-SIBLING Attribute 2
PARENT Attribute
PFCOLOR Attribute 3
PREV-SIBLING Attribute 2
PRIVATE-DATA Attribute
RESIZABLE Attribute 1
ROW Attribute
SELECTABLE Attribute 1
SELECTED Attribute
SENSITIVE Attribute
TABLE Attribute 2
TOOLTIP Attribute 1,4
TYPE Attribute 2
VISIBLE Attribute
WIDTH-CHARS Attribute
WIDTH-PIXELS Attribute
WINDOW Attribute 2
X Attribute
Y Attribute

1 Graphical interfaces only.

2 Readable only.

3 Character interfaces only.

4 Windows only.

Methods

LOAD-MOUSE-POINTER () Method

MOVE-TO-BOTTOM() Method

MOVE-TO-TOP() Method

LOAD-MOUSE-POINTER () Method (Graphical interfaces only)

Specifies the mouse pointer to display when the pointer is moved over the widget. If you apply this method to a frame, field group, or window, the same mouse pointer is displayed when it is moved across all child widgets within the frame, field group, or window. However, if you load a different mouse pointer for a child widget, the child widget mouse pointer is displayed when it is moved over that child.

Return Type: LOGICAL

Applies To: Browse, Browse-column, Button, Combo-box, Dialog-box, Editor, Field-group, Fill-in, Frame, Menu, Menu-item, Radio-set, Selection-list, Slider, Sub-menu, Toggle-box, Window

SYNTAX

LOAD-MOUSE-POINTER (*pointer-name*)
pointer-name

A character-string expression that specifies the name of a mouse pointer. Progress provides a collection of mouse pointers that you can use in graphical applications.

Table 56 names and describes each mouse pointer in the collection.

Pointer	Name	Description
APPSTARTING	Arrow	with an hourglass beside it
ARROW	Standard	arrow cursor
CROSS	Cross	hairs
HELP	Arrow	with a question mark beside it
IBEAM	I-beam	text cursor
NO	Circle	with a slash through it
RECTANGLE	(NT 3.51 only)	White rectangle
SIZE	Sizing	rectangle
SIZE-E	Size	to right
SIZE-N	Size	to top
SIZE-NE	Size	to top right
SIZE-NW	Size	to top left
SIZE-S	Size	to bottom
SIZE-SE	Size	to bottom right
SIZE-SW	Size	to bottom left
SIZE-W	Size	to left
UPARROW	Up	arrow
WAIT	System	busy
GLOVE	Glove/finger	
COMPILER-WAIT	Compiler	busy

NOTES

- If the mouse pointer is loaded successfully, the method returns TRUE.
- In addition to the mouse pointers that Progress supplies, you can also use a bitmap that you supply that is in the form of a Windows cursor (.cur or .ani) file. To use such a bitmap, substitute the name of the Windows cursor file for *pointer-name*.
- For browse-columns, if you do not specify a mouse pointer, Progress uses the mouse pointer the user specified for the browse.
- On Windows, you can specify a URL pathname. If you specify a fully-qualified URL, LOAD-MOUSE-POINTER loads the pointer file directly without searching directories or URLs in PROPATH. Valid URL protocols include HTTP and HTTPS.
- NOTE:** URL pathnames cannot contain the percent symbol (%). If an error exists in a URL specified on the PROPATH, the LOAD-MOUSE-POINTER method continues searching with the next PROPATH entry.
- If you specify URL pathnames on the PROPATH and your application repeatedly uses the LOAD-MOUSE-POINTER method with a URL pathname, you can improve performance by using the SEARCH function once to determine the full URL pathname to the directory containing the pointer files. Use this value to create a fully-qualified URL pathname for *pointer-name* and avoid repeated searches of the PROPATH.

MOVE-TO-TOP() Method

Moves the widget to the top (or front) of other widgets of the same class on the display.

Return Type: LOGICAL

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

SYNTAX

MOVE-TO-TOP ()

For the purposes of this method, the classes are as follows:

- Windows
- Frames
- Images and rectangles
- All other field-level widgets

If the operation is successful, the method returns TRUE.

Images and rectangles are displayed behind other field-level widgets and cannot be moved on

top of them.

When you use this method, set the KEEP-FRAME-Z-ORDER attribute to TRUE.

NOTE: In character interfaces, the MOVE-TO-TOP method applies only to the Frame.

MOVE-TO-BOTTOM() Method

Moves the widget to the bottom (or back) of other widgets of the same class on the display.

Return Type: LOGICAL

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

SYNTAX

MOVE-TO-BOTTOM ()

For the purposes of this method, the classes are as follows:

- Windows
- Frames
- Images and rectangles
- All other field-level widgets

If the operation is successful, the method returns TRUE.

When you use this method, set the KEEP-FRAME-Z-ORDER attribute to TRUE.

NOTE: In character interfaces, the MOVE-TO-BOTTOM method applies only to the Frame.

EVENTS

Type	Supported
Universal Key Function Events	-
Navigation Key Function Events	-
Field Editing Key Function Events	-
Default Keyboard Events	-
Mouse Events	Yes
General Direct Manipulation Events	Yes
Frame-only Direct Manipulation Events	-
Developer Events	Yes

Mouse Events

Progress supports two types of mouse events — portable and three-button events. You can use portable mouse events to associate triggers with logical actions of any mouse. You can use the three-button mouse events to associate triggers with specific physical actions of a three-button mouse.

The following tables reference portable mouse buttons for portable mouse events and physical mouse buttons for three-button mouse events. For more information on the mapping between portable and physical mouse buttons and how Progress processes mouse events in the 4GL, see the chapter on handling user input in the *Progress Programming Handbook*.

General Direct Manipulation Events

START-MOVE
START-RESIZE
START-ROW-RESIZE

Developer Events

Progress provides ten events, labeled U1 through U10, that you can invoke on any widget using the APPLY statement. The only function of a developer event is the one provided by your own trigger definition.

BGCOLOR Attribute (Graphical interfaces only)

Specifies the color number for the background color of the widget.

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Browse, Browse cell, Button, Combo-box, Control-frame, Dialog-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list,

Slider, Text, Toggle-box, Window

The color number represents an entry in the color table maintained by the COLOR-TABLE handle.

For a rectangle, if the FILLED attribute is TRUE, BGCOLOR specifies the color of the region inside the border of the rectangle.

For a browse cell, BGCOLOR specifies the color of a specific cell in the view port. You can set this color only as the cell appears in the view port during a ROW-DISPLAY event.

COLUMN Attribute

The column position of the left edge of the widget or the column position of the mouse cursor for the last mouse event on the display.

Data Type: DECIMAL

Access: Readable/Writeable

Applies To: Browse, Browse column, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, LAST-EVENT System Handle, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

For browse cells, field groups, and the LAST-EVENT handle, it is readable only.

For all widgets except windows, the COLUMN attribute specifies the location, in character units, of the left edge of the widget relative to the left edge of its parent widget. For windows, the location is relative to the left edge of the display.

For browse columns, the COLUMN attribute returns the UNKNOWN (?) value if the column is hidden.

For control-frames, the COLUMN attribute maps to the Left property of the control-frame COM object (ActiveX control container).

For the LAST-EVENT handle, the COLUMN attribute specifies the column location, in character units, of the last mouse event relative to the left edge of the current frame.

This attribute is functionally equivalent to the X attribute.

DATA-TYPE Attribute

A character value that represents the data type of the field associated with the widget. For example, the DATA-TYPE attribute of a slider widget always returns the value "INTEGER" because slider widgets can only represent integers.

Data Type: CHARACTER

Access: Readable/Writeable

Applies To: Browse column, Buffer-field Object Handle, Combo-box, Editor, Fill-in, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

This attribute is writeable for combo-boxes (only before realization), fill-ins, and text widgets only. For combo-boxes, writing to this attribute makes the drop-down list empty.

You must define this attribute as "CHARACTER" for SIMPLE and DROP-DOWN combo-boxes.

For widgets like image or rectangle, where a data type has no meaning, the attribute returns "UNKNOWN".

The DATA-TYPE attribute is only writable for dynamic fill-ins before they are realized. This attribute is read only for static fill-ins.

DCOLOR Attribute (Character Interfaces only)

The color number for the display color of the widget in character mode. This attribute is ignored in graphical interfaces.

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Browse, Browse cell, Button, Combo-box, Dialog-box, Editor, Fill-in, Frame, Literal, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

For browse widgets, it is readable only. For a browse cell, it specifies the color of a specific cell in the view port. You can set this color only as the cell appears in the view port during a ROW-DISPLAY event.

For rectangles, DCOLOR specifies the fill color. For windows, DCOLOR specifies the color

inherited by the menu bar, if the menu bar has no color specified.

The color number represents an entry in the color table maintained by the COLOR-TABLE handle.

You can now change the color of the background of menu frames (including menubars, submenus and pop-up menus) using the DCOLOR attribute. Previously, specifying the DCOLOR attribute for menus only changed the default color for menu items. Now, the DCOLOR attribute will be applied to the menu frame also. Note that no syntax changes were made. You can still specify the DCOLOR attribute for individual menu items.

For more information on widget color, see the [PFCOLOR Attribute](#).

DYNAMIC Attribute

Indicates whether the widget is dynamic or static.

Data Type: LOGICAL

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

If the DYNAMIC attribute is TRUE, the widget is dynamic, otherwise it is static.

EDGE-CHARS Attribute

The width, in character units, of the edge of a rectangle.

Data Type: DECIMAL

Access: Readable/Writeable

Applies To: Rectangle

EDGE-PIXELS Attribute

The width, in pixels, of the edge of a rectangle.

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Rectangle

FGCOLOR Attribute (Graphical interfaces only)

The color number for the foreground color of the widget.

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Browse, Browse cell, Button, Combo box, Dialog-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

The color number represents an entry in the color table maintained by the COLOR-TABLE handle.

For a browse cell, it specifies the color of a specific cell in the view port. You can set this color only as the cell appears in the view port during a ROW-DISPLAY event.

NOTE: This attribute has no meaning for control-frames because the ActiveX control visualization constitutes the foreground.

FILLED Attribute

Indicates whether to set the background color of a rectangle to a certain value.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Rectangle

If the value of the FILLED attribute is TRUE, the background color of the rectangle depends on the value of the BGCOLOR attribute (for graphical interfaces) or the value of the DCOLOR attribute (for character interfaces). The default value of FILLED is TRUE.

FRAME Attribute

The handle of the frame that contains the widget.

Data Type: WIDGET-HANDLE

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

This attribute is writeable only for static frames and all dynamic widgets. You can set this attribute for a static frame only before the widget is realized.

FRAME-COL Attribute

The decimal column position, in character units, of the left edge of the widget relative to the upper left corner of the frame that contains the widget.

Data Type: DECIMAL

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

FRAME-NAME Attribute

The name of the frame that contains the widget.

Data Type: CHARACTER

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

FRAME-ROW Attribute

The decimal row position, in character units, of the top edge of the widget relative to the upper left corner of the frame that contains the widget.

Data Type: DECIMAL

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

If the parent frame is a down frame with multiple occurrences, the FRAME-ROW attribute regards the original occurrence as the parent, not the current occurrence.

FRAME-X Attribute

The location of the left edge of the widget relative to the upper left corner of the frame that contains the widget.

Data Type: INTEGER

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

In character mode, this attribute returns the widget location in row column units. In graphical interfaces, this attribute returns pixels.

FRAME-Y Attribute

The location of the top edge of the widget relative to the upper left corner of the frame that contains the widget.

Data Type: INTEGER

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Editor, Fill-in, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

In character mode, this attribute returns the widget location in row column units. In graphical interfaces, this attribute returns pixels.

GRAPHIC-EDGE Attribute (Character interfaces only)

Indicates whether to draw a rectangle with graphic characters.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Rectangle

When the GRAPHIC-EDGE attribute is TRUE, the rectangle is drawn with graphic characters,

and the EDGE-CHARS and EDGE-PIXELS attributes are ignored.

HANDLE Attribute

A handle to the object.

Data Type: WIDGET-HANDLE

Access: Readable

Applies To: Browse, Browse cell, Buffer Object Handle, Buffer-field Object Handle, Button, Combo-box, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Query Object Handle, Radio-set, Rectangle, Selection-list, Slider, Server-socket Object Handle, Socket Object Handle, Sub-menu, Text, Toggle-box, Window

You can store this value in a WIDGET-HANDLE variable. You can also use it to associate one widget with another widget or with a system handle. For example, you can assign the HANDLE value of the menu bar to the MENU-BAR attribute of a window, or you can make the window the current window by assigning its HANDLE value to the CURRENT-WINDOW handle.

HEIGHT-CHARS Attribute (Graphical interfaces only)

The height, in character units, of the widget. The HEIGHT-CHARS attribute of the SESSION handle contains the height of the display.

Data Type: DECIMAL

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, SESSION System Handle, Slider, Text, Toggle-box, Window

For combo boxes, field groups, and the SESSION handle, this attribute is read-only.

For control-frames, the HEIGHT-CHARS attribute maps to the Height property of the control-frame COM object (ActiveX control container).

For browses, the HEIGHT-CHARS attribute sets the decimal height, in characters, of the browse without changing the height of the browse's rows. If you change the value of a browse's HEIGHT-CHARS or HEIGHT-PIXELS attribute, the number of rows that appear in the viewport might change.

HEIGHT-PIXELS Attribute (Graphical interfaces only)

The height, in pixels, of the widget. The HEIGHT-PIXEL attribute of the SESSION handle contains the height of the display.

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, SESSION System Handle, Slider, Text, Toggle-box, Window

For combo boxes, field groups, and the SESSION handle, this attribute is read-only.

For control-frames, the HEIGHT-PIXELS attribute maps to the Height property of the control-frame COM object (ActiveX control container).

For browses, the HEIGHT-PIXELS attribute sets the decimal height, in pixels, of the browse without changing the height of the browse's rows. If you change the value of a browse's HEIGHT-CHARS or HEIGHT-PIXELS attribute, the number of rows that appear in the viewport might change.

HELP Attribute

The help text for a field.

Data Type: CHARACTER

Access: Readable/Writeable

Applies To: Browse, Browse column, Buffer-field Object Handle, Button, Combo-box, Control-frame, Editor, Fill-in, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

For this attribute to have effect, the window that contains the specified widget must have its

STATUS-AREA attribute set to TRUE. The text stored in the HELP attribute displays in the status area of the containing window when the widget has input focus. The HELP attribute text overrides any status-area text issued by the STATUS statement.

HIDDEN Attribute

Indicates whether to “hide” a widget.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

Setting the HIDDEN attribute to TRUE prevents the widget from being displayed implicitly. For a field-level widget, child frame, or child window, this means that the widget is not automatically made visible when the containing frame or parent window becomes visible. The widget does not appear unless one of the following occurs:

- It is forced to receive user input (for example, using a SET or PAUSE statement).
- It is explicitly displayed using a VIEW statement or by setting its VISIBLE attribute to TRUE.

Any action that explicitly displays the widget also resets the HIDDEN attribute to FALSE. If the widget is already visible, setting its HIDDEN attribute to TRUE makes that widget and any widgets it parents (and their descendants) invisible (VISIBLE is set to FALSE). The default value of the HIDDEN attribute is FALSE for all widgets.

For windows, setting the HIDDEN attribute to TRUE prevents implicit display of the hidden window when you:

- Invoke DISPLAY, ENABLE, and VIEW statements for frames of the window.
- View an ancestor or descendant window of the hidden window.

This limits flashing side effects caused during set up of the application user interface. For windows, this attribute is not supported in character mode.

For frames and dialog boxes, setting the HIDDEN attribute to TRUE prevents implicit display of the frame or dialog box when you invoke DISPLAY or ENABLE statements for the widget or its descendant frames. This allows the frame or dialog box to remain invisible during actions that set it up. The HIDDEN attribute has no effect on DISPLAY statements directed to a file, pipe, or printer.

NOTE: Setting a frame or field-level widget’s VISIBLE attribute to TRUE also displays any parent or ancestor frames, even if their HIDDEN attributes are set to TRUE (resetting the HIDDEN attributes, if necessary). However, setting a window’s VISIBLE attribute to TRUE only displays the window if there are no ancestor windows with their HIDDEN attribute set to TRUE. In any case, the window’s own HIDDEN attribute is set to FALSE.

For field-level widgets and frames parented by other frames, setting the HIDDEN attribute to TRUE prevents implicit display of the field-level widget or child frame when its containing frame or dialog box is displayed. If the frame or dialog box containing the widget is visible, setting HIDDEN to FALSE for the widget makes the widget visible (the VISIBLE attribute is set to TRUE). If the containing frame or dialog box is not visible, setting HIDDEN to FALSE has no effect on the VISIBLE attribute of the widget.

NOTE: The HIDE statement sets the VISIBLE attribute for the widget to FALSE. It only sets the HIDDEN attribute to TRUE if you hide a field-level widget or child frame whose containing frame is still visible.

HWND Attribute

(Windows only; Graphical interfaces only)

An integer value for a Windows handle to the window that contains the widget.

Data Type: INTEGER

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

This attribute is supported for dynamic link library (DLL) access only on Windows. Some DLL routines require that you pass this value.

For Progress window widgets, the Windows window that contains the widget is actually the parent of the Windows widget referenced by HWND. Thus, to obtain the handle of the Windows window that contains the Progress window, you must pass the value of HWND to the GetParent() function (in the user32.dll). Pass the result of GetParent() to the DLL routine that requires it.

MANUAL-HIGHLIGHT Attribute

Indicates whether a widget exhibits custom or standard highlight behavior when selected.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Button, Combo-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

Set the MANUAL-HIGHLIGHT attribute to TRUE to use a customized highlight design for selection of the widget. A FALSE value for this attribute specifies the Progress default highlight behavior for the selection of the widget. For more information, see the [Progress Programming Handbook](#).

MOVABLE Attribute (Graphical interfaces only)

Indicates whether the widget can receive direct manipulation events.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Browse, Browse-column, Button, Combo-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

Set MOVABLE to TRUE to enable users to move the widget. To enable users to move more than one widget at a time, you must also set the SELECTABLE attribute to TRUE for each widget.

NOTE: Setting the MOVABLE attribute to TRUE enables direct manipulation events for the widget. These events take precedence over all other events. This effectively prevents data entry using the widget until all direct manipulation events are disabled (that is, until MOVABLE, RESIZABLE, and SELECTABLE are all FALSE).

NAME Attribute

A string identifier for the specified object or widget.

Data Type: CHARACTER

Access: Readable/Writeable

Applies To: Asynchronous Request Object Handle, Browse, Browse cell, Buffer Object Handle, Buffer-field Object Handle, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Query Object Handle, Radio-set, Rectangle, Selection-list, Server Object Handle, Server-socket Object Handle, Slider, Socket Object Handle, Sub-menu, Temp-table Object Handle, Text, Toggle-box, Window, X-document Object Handle, X-noderef Object Handle

For static data representation widgets, this attribute value defaults to the value of the name of the field or variable associated with the widget. You can use the NAME attribute to store any information associated with the widget.

For server handles, the NAME attribute defaults to the name of the Progress AppServer.

For control-frames, this attribute maps to the Name property of the of the control-frame COM object (ActiveX control container).

For dynamic widgets and asynchronous request handles, this attribute defaults to the unknown value (?).

CAUTION: If you change the value of this property at run time, any OCX event procedures that you have defined for a corresponding ActiveX control will not respond to control events because the events are sent with the new name.

For query objects, the NAME attribute applies only to static queries. For more information on query objects, see the [Progress Programming Handbook](#).

For temp-table objects, this returns the name of the temp-table as specified in the TEMP-TABLE-PREPARE method. If TEMP-TABLE-PREPARE() has not been called or the name has been cleared by CLEAR() and no subsequent TEMP-TABLE-PREPARE() has been

called, then this attribute returns the UNKNOWN value (?), which means that the table is in the UNPREPARED state.

For the X-document Object Handle or X-noderef Object Handle, this attribute returns the name of the XML node.

For any object or widget, this attribute can contain any arbitrary value that you set.

NEXT-SIBLING Attribute

The next entry in the list of asynchronous request handles, persistent procedure handles, server handles, temp-table object handles or widget handles, relative to a given handle.

Data Type: WIDGET-HANDLE

Access: Readable

Applies To: Asynchronous Request Object Handle, Browse, Buffer Object Handle, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu-item, Radio-set, Rectangle, Selection-list, Server Object Handle, Slider, Socket Object Handle, Server-socket Object Handle, Sub-menu, Text, THIS-PROCEDURE System Handle (and all procedure handles), Toggle-box, Window.

PARENT Attribute

The handle of the parent of a widget.

Data Type: WIDGET-HANDLE

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

This attribute is read-only for field groups.

For field-level widgets, the parent widget is the field group that contains the widget. For field groups, the parent widget is the frame that contains the field group. For frames, the parent widget is the window or field group that contains the frame. For windows, the parent widget is the window that parents the window. For a submenu or menu item, the parent widget is the menu or submenu that contains the submenu or menu item.

PFCOLOR Attribute (Character interfaces only)

The color number of the color of a widget that has input focus. The edge color of a rectangle widget.

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Browse cell, Button, Combo-box, Dialog-box, Editor, Fill-in, Frame, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Toggle-box, Window

The color number represents an entry in the color table maintained by the COLOR-TABLE handle.

For field-level widgets that receive focus, the PFCOLOR attribute specifies the input color for the widget. For windows, the PFCOLOR attribute specifies the color inherited by menu items in the menu bar when they are chosen, if the menu items don't already have the PFCOLOR specified.

For browse widgets, this color represents the input color for the focused cell.

For more information on widget color, see the [DCOLOR Attribute](#).

PREV-SIBLING Attribute

The previous entry in the list of persistent procedure handles, server handles, or widget handles, relative to a given handle.

Data Type: Handle

Access: Readable

Applies To: Asynchronous Request Object Handle, Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu-item, Radio-set, Rectangle, Selection-list, Server Object

Handle, Slider, Socket Object Handle, Server-socket Object Handle, Sub-menu, Text, THIS-PROCEDURE System Handle (and all procedure handles), Toggle-box, Window

PRIVATE-DATA Attribute

An arbitrary string associated with the handle of an object or widget.

Data Type: CHARACTER

Access: Readable/Writeable

Applies To: Asynchronous Request Object Handle, Browse, Browse-column, Buffer-field Object Handle, Buffer Object Handle, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Query Object Handle, Radio-set, Rectangle, Selection-list, Server Object Handle, Server-socket Object Handle, Slider, Socket Object Handle, Sub-menu, Text, THIS-PROCEDURE System Handle (and all procedure handles), Toggle-box, Window

Use this attribute any way you want. Progress does not check this attribute.

RESIZABLE Attribute (Graphical interfaces only)

Indicates whether the user can resize a widget at run time.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Browse, Browse-column, Button, Combo-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

If you set RESIZABLE to TRUE, the user can resize the widget. For the RESIZABLE attribute to take effect, you must also set the SELECTABLE attribute to TRUE.

NOTE: Setting the RESIZABLE attribute to TRUE enables direct manipulation events for the widget. These events take precedence over all other events. This effectively prevents data entry using the widget until all direct manipulation events are disabled (that is, until MOVABLE, RESIZABLE, and SELECTABLE are all FALSE).

ROW Attribute

The row position of the top edge of the widget relative to the top edge of the current iteration of a parent widget or the display. Specifies the row position of the mouse cursor for the last mouse event relative to the top edge of the display.

Data Type: DECIMAL

Access: Readable/Writeable

Applies To: Browse, Browse-column, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window, LAST-EVENT Handle

This attribute is read-only for browse columns, field groups, and the LAST-EVENT handle.

For all widgets except windows, the ROW attribute specifies the location, in character units, of the top edge of the widget relative to the top edge of its parent widget. For windows, the location is relative to the top edge of the display.

For a browse column, the ROW attribute returns the UNKNOWN (?) value if the column is hidden.

If the parent is a down frame with multiple occurrences, the ROW attribute regards the parent as the current occurrence.

For control-frames, the ROW attribute maps to the Top property of the control-frame COM object (ActiveX control container).

For the LAST-EVENT handle, the ROW attribute specifies the row location, in character units, of the last mouse event relative to the top edge of the current frame.

This attribute is functionally equivalent to the Y attribute.

SELECTABLE Attribute (Graphical interfaces only)

Indicates whether a widget is selectable for direct manipulation at run time.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

If the SELECTABLE attribute is TRUE, users can select and deselect the widget (that is, activate SELECTION and DESELECTION events for the widget). You must also set the SELECTABLE attribute to TRUE for the RESIZABLE attribute to take effect, allowing the user to resize the widget.

NOTE: Setting the SELECTABLE attribute to TRUE enables direct manipulation events for the widget. These events take precedence over all other events. This effectively prevents data entry using the widget until all direct manipulation events are disabled (that is, MOVABLE, RESIZABLE, and SELECTABLE are all set to FALSE). Also, vertical scrollbars are disabled until no direct manipulation can occur (that is, MOVABLE, RESIZABLE, and SELECTABLE are all set to FALSE)

SELECTED Attribute

Indicates whether a widget is selected (highlighted).

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Button, Combo-box, Editor, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

This attribute can be set in two ways — when the widget's SELECTABLE attribute is TRUE and the user selects the widget, or by setting the SELECTED attribute to TRUE from the 4GL whether or not its SELECTABLE attribute is TRUE. Although setting SELECTED to TRUE from the 4GL highlights the widget, this does not activate a SELECTION event for the widget.

SENSITIVE Attribute

Indicates whether a widget can receive input focus or events. Indicates whether certain Progress objects can receive events.

Data Type: LOGICAL

Access: Readable/Writeable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

Applies To: Server-socket Object Handle, Socket Object Handle

For widgets, if the SENSITIVE attribute is TRUE, the user can give input focus to the widget or can select, move, or resize it (if other attributes are set). A field-level widget must be specified in a frame before you can set the SENSITIVE attribute. The ENABLE statement implicitly sets this attribute to TRUE, and the DISABLE statement sets it to FALSE.

If the READ-ONLY attribute is TRUE for the widget, the SENSITIVE attribute has no effect except to grey out the widget in some environments.

For the socket and server socket objects, the SENSITIVE attribute indicates whether the object can receive events. The default value of this attribute is TRUE for socket and server socket objects.

If the SENSITIVE attribute is set to FALSE for the socket object, Progress will not execute the READ-RESPONSE procedure for the socket even if the READ-RESPONSE event occurs.

If the SENSITIVE attribute is set to FALSE for the server socket object, Progress will stop accepting connections on the port associated with the server socket.

NOTE: If an application knows it will not receive data on a socket during some period of time, it should set this attribute to FALSE. This allows the application to run more efficiently since Progress does not monitor the socket if its SENSITIVE attribute is set to FALSE. Data can still be written to an insensitive socket object. When the attribute is set to TRUE, Progress checks the socket for data.

TABLE Attribute

The name of the database table containing the field associated with a widget, buffer, or buffer-field.

Data Type: CHARACTER

Access: Readable

Applies To: Browse columnBuffer Object Handle, Buffer-field Object Handle,

Combo-box, Editor, Fill-in, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box

NOTE: The TABLE attribute of a buffer contains the name of the table, not the name of the buffer.

TOOLTIP Attribute (Windows only; Graphical interfaces only)

A help text message for a text field or text variable. Progress automatically displays this text when the user pauses the mouse pointer over a widget for which a tooltip is defined.

Data Type: CHARACTER

Access: Readable/Writable

Applies To: Browse, Button, Combo-box, Editor, Fill-in, Image, Radio-set, Rectangle, Selection-list, Slider, Text, and Toggle-box

You can add or change the TOOLTIP attribute at any time. If TOOLTIP is set to "" or ? (the unknown value), then the ToolTip is removed. No ToolTip is the default.

TYPE Attribute

The type of a handle.

Data Type: CHARACTER

Access: Readable

Applies To: Asynchronous Request Object Handle, Browse, Browse cell, Buffer Object Handle, Buffer-field Object Handle, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Query Object Handle, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window, CLIPBOARD System Handle, CODEBASE-LOCATOR System Handle, COLOR-TABLE System Handle, COMPILER System Handle, CURRENT-WINDOW System Handle, DEBUGGER System Handle, DEFAULT-WINDOW System Handle, ERROR-STATUS System Handle, FILE-INFO System Handle, FOCUS System Handle, FONT-TABLE System Handle, LAST-EVENT System Handle, RCODE-INFO System Handle, SELF System Handle, Server Object Handle, Server-socket Object Handle, SESSION System Handle, Socket Object Handle, THIS-PROCEDURE System Handle (and all procedure handles), X-document Object Handle, X-noderef Object Handle

The TYPE attribute returns the widget or handle type, for example, "WINDOW", "FRAME", "BUTTON," "MENU," or "SERVER."

If a system handle (such as CURRENT-WINDOW or FOCUS) refers to a user interface widget, the TYPE attribute returns the type of that widget. If a system handle (such as SESSION or CLIPBOARD) refers to a Progress status or system widget, the TYPE attribute value is "PSEUDO-WIDGET".

For procedure handles and system handles that refer to procedures (such as THIS-PROCEDURE), the TYPE attribute returns "PROCEDURE".

For an asynchronous request handle, the TYPE attribute returns "ASYNC-REQUEST".

For a server-socket handle, the TYPE attribute returns "SERVER-SOCKET", and for a socket handle, it returns "SOCKET".

For the X-document and X-noderef object handles, the TYPE attribute returns "X-DOCUMENT" and "X-NODEREF", respectively.

VISIBLE Attribute

Indicates whether a widget is currently visible on the display.

Data Type: LOGICAL

Access: Readable/Writable

Applies To: Browse, Browse-column, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

This attribute is read-only for field groups, menus, menu items, and submenus. A field-level widget must be specified in a frame definition before you set its VISIBLE attribute.

The behavior of the **VISIBLE** attribute depends on the setting of the **HIDDEN** attribute of related widgets:

- When you set the **VISIBLE** attribute of a window to **TRUE**:
 - Progress displays that window and the widgets it contains whose **VISIBLE** attributes are already set to **TRUE**. Otherwise, you must explicitly **DISPLAY** or **VIEW** a widget, or otherwise set a widget's **VISIBLE** attribute to **TRUE** in order to display it in the window.
 - Progress displays that window and all ancestor windows only if no ancestor window has its **HIDDEN** attribute set to **TRUE**. If Progress displays the window, it also displays all descendant windows down to, but not including, the first descendant window whose **HIDDEN** attribute is set to **TRUE**.
- When you set the **VISIBLE** attribute of any widget within a window to **TRUE**, Progress displays that widget, any ancestor frames, and the window (if necessary), unless the **HIDDEN** attribute of the window is **TRUE**. If the window's **HIDDEN** attribute is **TRUE**, Progress sets the **VISIBLE** attributes of the widget and any ancestor frames to **TRUE** and sets the **HIDDEN** attributes of the widget and its ancestor frames to **FALSE** without displaying them.
- When you set the **VISIBLE** attribute of a frame to **TRUE**, Progress displays all of its field-level widgets and descendant frames, except those whose **HIDDEN** attributes are **TRUE**.
- When you explicitly set the **VISIBLE** attribute of any widget to **TRUE**, Progress sets its **HIDDEN** attribute to **FALSE**. If you explicitly set the **VISIBLE** attribute of a field-level widget or child frame to **FALSE** while its parent frame remains visible, Progress also sets the **HIDDEN** attribute of the field-level widget or child frame to **TRUE**. If you explicitly set the **VISIBLE** attribute of a child window to **FALSE**, the **HIDDEN** attribute of the child window remains unchanged, whether or not the parent window is visible.
- The following behavior is true for the browse column:
 - The syntax of the **VISIBLE** attribute for the browse column is as follows:
 - The behavior of the **VISIBLE** attribute for a browse column does not depend on the setting of the **HIDDEN** attribute of the related widget.
 - Changing the **VISIBLE** attribute of a browse column may affect which columns are locked if **NUM-LOCKED-COLUMNS** has been set. This is because **NUM-LOCKED-COLUMNS** only applies to visible columns. For example, if the first three columns of a browse are locked and the second column is made not **VISIBLE**, the fourth column will then become locked.
 - If a widget is not already realized and you set its **VISIBLE** attribute to **TRUE**, Progress realizes that widget.
 - In character mode, the **VISIBLE** attribute is always set to **TRUE**.

WIDTH-CHARS Attribute **(Graphical interfaces only)**

The width of the widget or the display used in the current session, in character units.

Data Type: DECIMAL

Access: Readable/Writable

Applies To: Browse, Browse-column, Buffer-field Object Handle, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, SESSION System Handle, Slider, Text, Toggle-box, Window

The attribute is read-only for field groups, and the **SESSION** handle.

For control-frames, the **WIDTH-CHARS** attribute maps to the **Width** property of the control-frame **COM** object (ActiveX control container).

For editor widgets, this attribute can set the word wrap margin for the **WORD-WRAP** attribute.

For more information, see the [WORD-WRAP Attribute](#) reference entry.

For buffer-field objects, the **WIDTH-CHARS** attribute is the number of characters in the **STRING-VALUE**, which Progress calculates using the **FORMAT** attribute. In addition, the **WIDTH-CHARS** attribute of a buffer-field is readable but not writeable.

For browses, the **WIDTH-CHARS** attribute sets the width, in characters, of the browse without changing the width of any browse-column. If you change the value of a browse's **WIDTH-CHARS** or **WIDTH-PIXELS** attribute, the horizontal scrollbar might appear or

disappear, which might cause the number of rows that appear in the viewport to change. For browse-columns, the WIDTH-CHARS attribute sets the width, in characters, of the browse-column without changing the width of the browse.

WIDTH-PIXELS Attribute (Graphical interfaces only)

The width of the widget or the screen display used in the current session, in pixels.

Data Type: INTEGER

Access: Readable/Writable

Applies To: Browse, Browse-column, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Radio-set, Rectangle, Selection-list, SESSION System Handle, Slider, Text, Toggle-box, Window

The attribute is read-only for field groups, and the SESSION handle.

For control-frames, the WIDTH-PIXELS attribute maps to the Width property of the control-frame COM object (ActiveX control container).

For editor widgets, this attribute can set the word wrap margin for the WORD-WRAP attribute.

For more information, see the [WORD-WRAP Attribute](#) reference entry.

For browses, the WIDTH-PIXELS attribute sets the width, in pixels, of the browse without changing the width of any browse-column. If you change the value of a browse's

WIDTH-CHARS or WIDTH-PIXELS attribute, the horizontal scrollbar might appear or disappear, which might cause the number of rows that appear in the viewport to change.

For browse-columns, the WIDTH-PIXELS attribute sets the width, in pixels, of the browse-column without changing the width of the browse.

WINDOW Attribute

A handle to the window that owns a widget or that contains the owner of a widget.

Data Type: WIDGET-HANDLE

Access: Readable

Applies To: Browse, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, Literal, Menu, Menu-item, Radio-set, Rectangle, Selection-list, Slider, Sub-menu, Text, Toggle-box, Window

For a menu bar or pop-up menu of a window, the WINDOW and OWNER attributes have the same value.

In the case of a window, the WINDOW attribute returns the window's widget-handle (not its parent's handle, if any). For a menu bar or pop-up menu of a window, the WINDOW and OWNER attributes have the same value.

X Attribute

The pixel location of the left edge of a widget relative to the left edge of the parent widget or the display. The pixel location of the mouse cursor relative to the left edge of the display (for the last mouse event).

Data Type: INTEGER

Access: Readable/Writable

Applies To: Browse, Browse cell, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, LAST-EVENT System Handle, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

This attribute is read-only for field groups, browse cells, and the LAST-EVENT handle.

For all user interface widgets except windows, the X attribute specifies the location, in pixels, of the left edge of the widget relative to the left edge of its parent widget. For windows, it is the location of the left edge of the window relative to the left edge of the display.

For a browse column, the X attribute returns the UNKNOWN (?) value if the column is hidden.

For control-frames, the X attribute maps to the Left property of the control-frame COM object (ActiveX control container).

For the LAST-EVENT handle, the X attribute returns the pixel location of a mouse event relative to the left edge of the current frame.

This attribute is functionally equivalent to the COLUMN attribute.

Y Attribute

The pixel location of the top edge of the widget relative to the top edge of the parent widget or the display. The pixel location of the mouse cursor relative to the top edge of the display (for the last mouse event).

Data Type: INTEGER

Access: Readable/Writeable

Applies To: Browse, Browse cell, Button, Combo-box, Control-frame, Dialog-box, Editor, Field-group, Fill-in, Frame, Image, LAST-EVENT System Handle, Literal, Radio-set, Rectangle, Selection-list, Slider, Text, Toggle-box, Window

This attribute is read-only for field groups, browse cells, and the LAST-EVENT handle.

For all user interface widgets except windows, the Y attribute specifies the location, in pixels, of the top edge of the widget relative to the top edge of its parent widget. For windows, it is the location of the top edge of the window relative to the top edge of the display.

For a browse column, the Y attribute returns the UNKNOWN (?) value if the column is hidden.

For control-frames, the Y attribute maps to the Top property of the control-frame COM object (ActiveX control container).

For the LAST-EVENT handle, the Y attribute returns the pixel location of a mouse event relative to the top edge of the current frame.

This attribute is functionally equivalent to the ROW attribute.