

isCOBOL Evolve: GUI Reference Guide

Key Topics:

- [Working With Controls](#)
- [Controls Reference](#)
- [Character Based Screens](#)



Copyrights

Copyright (c) 2019 Veryant
6390 Greenwich Drive, #225, San Diego, CA 92122, USA

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and recompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Veryant and its licensors, if any.

Chapter 1

Working With Controls

isCOBOL supports all important graphical controls for realizing a complete User Interface. Please refer to the isCOBOL Appendices manual, [Graphical Control List](#), to display the complete list of controls with its properties, styles and events

isCOBOL supports the following graphical controls:

- [BAR](#)
- [BITMAP](#)
- [CHECK-BOX](#)
- [COMBO-BOX](#)
- [DATE-ENTRY](#)
- [ENTRY-FIELD](#)
- [FRAME](#)
- [GRID](#)
- [JAVA-BEAN](#)
- [LABEL](#)
- [LIST-BOX](#)
- [PUSH-BUTTON](#)
- [RADIO-BUTTON](#)
- [SCROLL-BAR](#)
- [SLIDER](#)
- [STATUS-BAR](#)
- [TAB-CONTROL](#)
- [TOOL-BAR](#) *
- [TREE-VIEW](#)
- [WEB-BROWSER](#)
- [WINDOW](#) *

* TOOL-BAR and WINDOW can be used only in conjunction with the DISPLAY Statement.

Properties, Styles and Events

Controls have properties and styles that affect their look and behavior. While styles do not take any value and cannot be inquired, properties always have a value that can be set and inquired.

Controls raise events when the user interacts with them. Events can be captured and handled by the program through Event and [Embedded Procedures](#).

Creating a control

Displaying single controls

The only way to create a control is to use the DISPLAY statement.

```
DISPLAY control-class

{ { property-name           } [IS ] property-option [GIVING result-1] } ...
{ PROPERTY property-type   } [ARE]
                               [= ]

{ style-name } ...
{ event-procedure }
[ HANDLE [IN] control-handle ]
```

control-class is one of the supported controls. See the list at the top of this page.

property-option can be one of the following:

```
{ property-value [LENGTH {IS} length-1 ] }
                               {= }
{ ( {property-value} ... ) }
{ [MULTIPLE] property-table }
{ [TABLE ] {= } }
```

control-handle is the handle that can be used to modify, inquire or destroy that control. If the HANDLE phrase is not specified, the control can be referenced only indicating its position. This technique is deprecated.

Each time such DISPLAY is executed, a new control is created. If it is displayed at the same location as an existing control with the TEMPORARY Style set, the existing control is destroyed first. It may not be easy to notice that the program is displaying unneeded controls because they may overlap exactly, however, performance will decrease dramatically without any apparent reason. Use the [MODIFY](#) Statement to update already displayed controls.

Displaying Screen Section items

A control can also be defined as part of a Screen Section item:

```
level-number [control-handle] control-class

{ property-name [IS] property-value }...

{ style-name } ...

{ embedded-procedure }
{ event-procedure }

.
```

embedded-procedure is defined as stated below. Refer to the [Embedded Procedures](#) section for a more detailed explanation.

```
[ BEFORE PROCEDURE IS procedure-1 [{THROUGH} procedure-2] ]
                                {THRU  }

[ AFTER PROCEDURE IS procedure-3 [{THROUGH} procedure-4] ]
                                {THRU  }

[ EXCEPTION PROCEDURE IS procedure-5 [{THROUGH} procedure-6] ]
                                {THRU  }
```

event-procedure is defined as stated below. Refer to the [Event handling](#) section for a more detailed explanation.

```
[ EVENT PROCEDURE IS procedure-1 [{THROUGH} procedure-2] ]
                                {THRU  }
```

A Screen Section item can also consist of a group containing controls. This is useful to create or destroy several controls with a single statement. This practice grants better performance, in comparison with the creation or destruction of one control at a time.

Below, is a sample of how a Screen Section item looks like:

```
01 MyScreen.
  03 MyLabel LABEL
      LINE          2
      COL           2
      SIZE          10 CELLS
      TITLE         "This is a label"
      .
  03 MyEntryField ENTRY-FIELD
      LINE          2
      COL           13
      SIZE          20 CELLS
      VALUE         AnyDataItem
      .
```

MyScreen is the handle of the whole Screen Section item. *MyLabel* and *MyEntryField* are the handles of the controls.

The following statement creates the label and the entry-field at the position specified above and sets all of their properties:

```
DISPLAY MyScreen
```

If the same statement is executed again, all the properties are updated. An internal optimizer ensures that the properties are updated only when needed.

If the program displays a single item that has already been created, only its properties are updated. The optimizer works with single item DISPLAYs, too. The following statement updates only the properties defined for the Entry-Field identified by the handle *MyEntryField*.

```
DISPLAY MyEntryField
```

Modifying a control

The MODIFY Statement can be used to change a single property or to set or reset a style. Generally, the best way to interact with a control is to change only the properties that need to be changed.

The syntax of the MODIFY Statement is:

Format 1

```
MODIFY {control-item      } [ ( {index-1} ... ) ]  
      {CONTROL AT location}  
  
      { property-name      } [IS ]      { property-option [GIVING result-1] } ...  
      { PROPERTY property-type } [ARE]  
                                   [= ]  
  
      { [NOT] style-name } ...  
  
      { event-procedure }
```

location is defined as follows:

```
{ screen-loc [CELL ]  
              [CELLS ]  
              [PIXEL ]  
              [PIXELS]  
  
  LINE NUMBER line-num [CELL ]  
                      [CELLS ]  
                      [PIXEL ]  
                      [PIXELS]  
  
  {COLUMN } NUMBER col-num [CELL ]  
  {COL     }                [CELLS ]  
  {POSITION}                [PIXEL ]  
  {POS     }                [PIXELS]  
  
}
```

property-option can be one of the following:

```
{ property-value [LENGTH {IS} length-1 ] }
                    {= }
{ ( {property-value} ... ) }
{ [MULTIPLE] property-table }
{ [TABLE ] {= } }
```

Format 2

```
MODIFY {window-handle }
      {WINDOW [generic-handle] }

{ property-name [IS ] property-value }
{           [= ]           } ...

{ [NOT] style-name } ...

[ ON EXCEPTION statement-1]

[ NOT ON EXCEPTION statement-2 ]

{ event-procedure }

[END-MODIFY]
```

When more than one property or style are changed with the same MODIFY Statement, the changes occur in the exact order they are written. The statement

```
MODIFY MyEntryField, ENABLED = 1, VALUE = "New value"
```

produces the same effect than the following distinct statements:

```
MODIFY MyEntryField, ENABLED = 1
MODIFY MyEntryField, VALUE = "New value"
```

Generally speaking, we recommend the use of the MODIFY Statement, unless a lot of controls are to be changed at the same time. If your program uses a number of controls to display the content of a record and you load a new record, you should update the screen displaying the whole Screen Section that defines all the controls. Modifying every single control may be slower.

Inquiring a control

When the program needs to retrieve the value of a property, it must use the INQUIRE Statement

In order to make programs easier, the property VALUE is automatically updated, as long as it has been declared in Screen Section. Therefore, it is never needed to inquire the VALUE property, except during the event handling, because events may be asynchronous and the variable associated with the VALUE property may not be updated at the time the program needs it.

The syntax of the INQUIRE Statement is:

Format 1

```
INQUIRE {control-item          } [ ( {index-1} ... ) ]
        {CONTROL AT location}

{ { property-name              } [IN] { [MULTIPLE] property-value [LENGTH {IN} length-
1 ] } } ...
{ PROPERTY property-
type }          { [TABLE   ]                { = } }
```

location is defined as follows:

```
{ screen-loc [CELL  ]
              [CELLS ]
              [PIXEL ]
              [PIXELS]

LINE NUMBER line-num [CELL  ]
                  [CELLS ]
                  [PIXEL ]
                  [PIXELS]

{COLUMN } NUMBER col-num [CELL  ]
{COL    }                [CELLS ]
{POSITION}                [PIXEL ]
{POS    }                [PIXELS]

}
```

Format 2

```
INQUIRE {window-handle          }
        {WINDOW [generic-handle] }
{ property-name [ IN ] property-value }
{              [ = ]                } ...
[END-INQUIRE]
```


property-name can be one of the following:

- COLUMN
- HINT
- LINE
- LINES
- POP-UP MENU
- SCREEN COLUMN
- SCREEN LINE
- SIZE
- TITLE
- VISIBLE

The program can retrieve values of several properties with the same INQUIRE command:

```
INQUIRE MyEntryField ENABLED IN EnabledVar, VALUE IN ValueVar.
```

produces the same effect as the following distinct statements:

```
INQUIRE MyEntryField ENABLED IN EnabledVar.  
INQUIRE MyEntryField VALUE IN ValueVar.
```

Destroying a control

When a control is no longer necessary, it should be destroyed. All the resources used by that control are released.

The syntax of the DESTROY Statement is:

Format 1

```
DESTROY { screen-name-1      } ...  
        { handle-1           }  
        { CONTROL AT location }
```

location is defined as follows:

```
{ screen-loc [CELL  ]
               [CELLS ]
               [PIXEL ]
               [PIXELS]

  LINE NUMBER line-num [CELL  ]
                      [CELLS ]
                      [PIXEL ]
                      [PIXELS]

  {COLUMN  } NUMBER col-num [CELL  ]
  {COL      }                [CELLS ]
  {POSITION}                [PIXEL ]
  {POS      }                [PIXELS]

}
```

Format 2

```
DESTROY ALL CONTROLS
```

Format 3

```
DESTROY {window-handle      }
```

Destroying every single control may be slow, the best approach is to destroy the whole Screen in which controls are defined. When a graphical window is destroyed, all controls associates to the window are automatically destroyed. The main application Window can't be destroyed.

Activating a control

Controls are activated by the ACCEPT Statement. The syntax is:

```
ACCEPT {screen-name-1      } ...
       {handle-1           }
       {CONTROL AT location}
```

location is defined as follows:

```
{ screen-loc [CELL  ]
               [CELLS ]
               [PIXEL ]
               [PIXELS]

  LINE NUMBER line-num [CELL  ]
                       [CELLS ]
                       [PIXEL ]
                       [PIXELS]

  {COLUMN  } NUMBER col-num [CELL  ]
  {COL      }                [CELLS ]
  {POSITION}                [PIXEL ]
  {POS      }                [PIXELS]

}
```

When a whole Screen Section is activated, only one control is activated at a time and the other controls are automatically activated in response to the user's actions. By default, the [Tab] and [Enter] keys activate the next control, while the [Shift+Tab] key activates the previous one. The keyboard is fully configurable, so any key can be used to change the active control. Controls are also activated by the mouse click. The program can prevent a control from being activated. See the [Embedded Procedures](#) and the [Event handling](#) sections for further details.

When a single control is activated, no automatic activation occurs. The CMD-GOTO Event is fired instead, and all the information needed to activate the new control is provided.

The user can also use the function keys to access the various functions of the program. The easiest way to do that, is to include the ACCEPT Statement in a loop.

```
77 Key-Pressed IS SPECIAL-NAMES CRT STATUS PIC 9(5) .
78 KeyEsc VALUE 27 .
78 KeyF6 VALUE 6 .

.
.
.

PERFORM UNTIL Key-Pressed = KeyEsc

  ACCEPT MyScreen

  EVALUATE Key-Pressed
  WHEN KeyF6
    PERFORM A-FUNCTION
  END-EVALUATE

END-PERFORM
```

Key-Pressed is defined as the special register CRT STATUS containing the value associated with the function key that terminated the ACCEPT Statement.

Push-Buttons, Check-Boxes and Radio-Buttons with the SELF-ACT Style set do not need to be explicitly activated. In this way, a control not belonging to the Screen Section currently active can be handled with little programming effort.

The STYLE common property

STYLE is a common property that holds a numeric value. This value is the sum of the numeric values of the individual styles that have been applied to a particular control. Styles affect the appearance or behavior of a control. For example, some of the styles that apply to a button include: BITMAP, FRAMED, and NOTIFY. Individual styles have a predefined numeric value assigned in the file [iscontrols.def](#) and do not take any other value. A style can be applied or not applied. If the style is indicated in a statement, it is applied to the control; if it is absent, it is not applied. Most styles pertain to only a certain type of control, although a few are common to all controls.

There are two ways to specify a style for a control:

- Include the style name in the statement that creates the control
- Add the style's numeric value to any other style values that apply, and include the sum value in the STYLE IS style-flags phrase, where style-flags is the sum value

Including the style name in the statement that creates the control is the usual method for specifying a style. The collection of style names included in the statement (such as BITMAP and NOTIFY) instructs the Compiler to build the appropriate STYLE property value (the STYLE property is technically the method by which all styles are stored in the run-time system). However, there is an important restriction: the Compiler understands style names only when it knows what kind of control is being built. If you specify a control that has a variable (undefined) type, then you must specify styles with the *STYLE IS style-flags* phrase. In this case, you construct the STYLE property value by adding together the appropriate style numbers. Each style has a corresponding numeric value, and the STYLE property holds the sum of the specified styles. For example, if the numbers corresponding to BITMAP and NOTIFY were 1 and 4 respectively, the phrase STYLE IS 5 would specify those two styles. Each style's identifying number can be found in the file [iscontrols.def](#) (they are level 78 items).

You can use the MODIFY statement to change a style value after a control has been created. However, in many cases the style change may not take effect. Exactly which styles can be effectively changed for each control is not known and is, therefore, not documented. When a style is known to be modifiable or not modifiable, that information is documented with the style. We recommend, however, that you test the behavior of your application in the target environment to confirm that style changes are handled in the way that you expect

Embedded Procedures

Embedded Procedures have been implemented to make programming even simpler. It is a way to attach a procedure, a paragraph, or a section to one or more controls and have them activated before the control activation, after the control activation, and when a function key is pressed.

Every Screen Section item, regardless of whether it is a group or a single control, can handle up to three different Embedded Procedures. If an Embedded Procedure is assigned to a group, it is inherited by all the groups or controls that it contains, unless they have their own embedded procedure set.

There are three kinds of Embedded Procedure:

```
[ BEFORE PROCEDURE IS procedure-1 [{THROUGH} procedure-2] ]
                                {THRU }
```

When BEFORE PROCEDURE is declared for the control, the paragraph(s) procedure-1 (thru procedure-2) are executed when the control gets focus, just before accepting user input.

```
[ AFTER PROCEDURE IS procedure-3 [{THROUGH} procedure-4] ]
                                {THRU  }
```

When AFTER PROCEDURE is declared for the control, the paragraph(s) procedure-3 (thru procedure-4) are executed when the control loses focus or when the ACCEPT terminates, just after accepting user input.

```
[ EXCEPTION PROCEDURE IS procedure-5 [{THROUGH} procedure-6] ]
                                {THRU  }
```

When EXCEPTION PROCEDURE is declared for the control, the paragraph(s) procedure-5 (thru procedure-6) are executed each time the ACCEPT of user input is interrupted by an exception (for example when the user presses a function key).

Within Embedded Procedures, the [SCREEN CONTROL](#) special registry can be used to monitor and change the focus.

Event handling

As explained above, Embedded Procedures can be used to handle the most common actions the user can do, such as activating and leaving a control.

The user can interact with controls in a number of ways, however, and creating a new Embedded Procedure for every possible action would be confusing and not efficient at all. Handling events is preferable in most cases.

Events are categorized into three groups:

- **command events** (whose name begins with "CMD-") correspond to actions taken by the user that the program needs to act on, such as closing a window or pushing a button. When a command event occurs, the runtime system assigns a value to the EVENT STATUS and then terminates the current ACCEPT with an exception value of "96"
- **notify events** (whose name begins with "NTF-") correspond to informational events that the program may not have to act on, such as editing a text-field or resizing the window. When a notify event occurs, the runtime system assigns a value to the EVENT STATUS and then terminates the current ACCEPT with an exception value of "96"
- **messages** (whose name begins with "MSG-") pass information to a screen control's Event Procedure. This division is somewhat arbitrary, but corresponds to the most common situations. Messages are different from other events, because they do not terminate the current ACCEPT. Messages are sent only to a control's Event Procedure.

Event names are listed as constant items in the [isgui.def](#) copybook.

Just one Event Procedure can be assigned to a control at a time and it is executed each time that control fires an event. Within Event Procedures, the [EVENT STATUS](#) special registry can be used to monitor and change the event behavior.

Events cannot be nested, therefore in the event procedure code you shouldn't:

- Perform other ACCEPTs of user input (i.e. call another program that opens a new window and performs an ACCEPT on it).
- Perform actions that generate events (i.e. use the grid [Action](#) property inside grid events).

For the above needs, it's suggested that you:

- Set a flag variable and make the event terminate the ACCEPT.

```
TREEVIEW-EVENTS.  
  if event-type = msg-tv-dblclick  
    move 1 to flag-call  
    set event-action to event-action-terminate  
  end-if.
```

- Test the flag and perform the proper action after the ACCEPT is terminated.

```
accept Screen1  
  on exception  
    |do something here  
end-accept  
if flag-call = 1  
  move 0 to flag-call  
  call "ProgramWithNewWindow"  
end-if
```

Note: Embedded and Event procedures are paragraphs and sections automatically executed by the runtime while the user interacts with the screen. The program jumps to these paragraphs as if a PERFORM statement was issued, then, when the paragraph code has been executed, the program returns to the ACCEPT statement. Therefore, it is strongly suggested that you avoid using GO TO statements into these paragraphs; if the program jumps outside these paragraphs through a GO TO statement, it may not be able to return to the ACCEPT, causing it to hang.

Performance Tuning: In the Thin Client environment, when the focus changes, no information is sent from the client to the server if:

- the control doesn't have embedded or event procedures
- the control doesn't format its value on exit (for example: numeric field with decimal or edit type)
- the focus change doesn't terminate the ACCEPT with TERMINATION or EXCEPTION

In this case the program will run faster.

HTML and Controls

Unless explicitly excluded, display-only texts can be HTML source. This makes it easy to create fancy interfaces.

The HTML text must be enclosed between the <HTML> and </HTML> tags. Fonts and colors can be mixed. The following label

```
LABEL TITLE '<html><font size=6>f</font><font style="background-  
color:#FFFF00"><font size=5>o</font><font size=4>n</font><font size=3>t s</  
font><font size=4>i</font><font size=5>z</font></font><font size=6>e</font></  
html>'
```

is rendered as follows:

font size

Images, tables and borders are supported, too.

Within HTML titles the "&" character is interpreted as the beginning of an HTML entity (e.g. * *;) while the first letter underlined by the <U> tag becomes the access key.

Color management

There are two ways of managing colors.

1. Using standard COBOL values
2. Using RGB

Using standard COBOL values

The first way, provided for compatibility reasons, uses up to sixteen values, numbered from 1 to 16. Values 1 to 8 are base colors, 9 to 16 are their brighter version. The file "[iscobol.def](#)" contains the color definitions. They are divided into groups, and can be combined:

ForegroundColor	78	black	value 1.
	78	blue	value 2.
	78	green	value 3.
	78	cyan	value 4.
	78	red	value 5.
	78	magenta	value 6.
	78	brown	value 7.
	78	white	value 8.
	78	dark-gray	value 9.
	78	bright-blue	value 10.
	78	bright-green	value 11.
	78	bright-cyan	value 12.
	78	bright-red	value 13.
	78	bright-magenta	value 14.
	78	yellow	value 15.
	78	bright-white	value 16.
ForegroundBrightness	78	frgrnd-low	value 2048.
	78	frgrnd-high	value 4096.

BackgroundColor	78	bckgrnd-black	value 32.
	78	bckgrnd-blue	value 64.
	78	bckgrnd-green	value 96.
	78	bckgrnd-cyan	value 128.
	78	bckgrnd-red	value 160.
	78	bckgrnd-magenta	value 192.
	78	bckgrnd-brown	value 224.
	78	bckgrnd-white	value 256.
	78	bckgrnd-dark-gray	value 288.
	78	bckgrnd-bright-blue	value 320.
	78	bckgrnd-bright-green	value 352.
	78	bckgrnd-bright-cyan	value 384.
	78	bckgrnd-bright-red	value 416.
	78	bckgrnd-bright-magenta	value 448.
	78	bckgrnd-yellow	value 480.
	78	bckgrnd-bright-white	value 512.
BackgroundBrightness	78	bckgrnd-low	value 65536.
	78	bckgrnd-high	value 131072.
GenericAttribute	78	color-reverse	value 1024.
	78	color-underline	value 8192.
	78	color-blink	value 16384.
	78	color-protected	value 32768.

When an color value is used with a property that defines both the foreground color and the background colors, the value is computed as follows:

```

Zero

[ + ForegroundColor]

[ + BackgroundBrightness]

[ + BackgroundColor]

[ + BackgroundBrightness]

[ + GenericAttribute ] ...

```


When an color value is used with a property that defines either the foreground color or the background color, the value can be only 0 to 15 and the corresponding color is applied to foreground or background. The table below shows the possible values for BACKGROUND-COLOR and FOREGROUND-COLOR properties.

0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Dark Gray
9	Bright Blue
10	Bright Green
11	Bright Cyan
12	Bright Red
13	Bright Magenta
14	Yellow
15	Bright White

Brightness can be also affected by the following clauses:

BACKGROUND-HIGH BACKGROUND-LOW BACKGROUND-STANDARD HIGHLIGHT LOWLIGHT STANDARD

When the REVERSE-VIDEO phrase is specified, background and foreground colors are swapped.

When the SAME phrase is specified, the whole screen item for which it is specified is displayed with the same colors and attributes of the screen position occupied by its first character.

Properties that define both the foreground color and the background color are:

- Cell-Color
- Cell-Current-Color
- Cell-Entry-Color
- Color
- Column-Color
- Cursor-Color
- Drag-Color
- Heading-Color
- Region-Color
- Row-Color
- Row-Color-Pattern
- Row-Cursor-Color

Properties that define only the background color are:

- Background-Color

- Cell-Background-Color
- Cell-Current-Background-Color
- Cell-Entry-Background-Color
- Column-Background-Color
- Cursor-Background-Color
- Decoration-Background
- Drag-Background-Color
- Heading-Background-Color
- Row-Background-Color
- Row-Background-Color-Pattern
- Row-Cursor-Background-Color

Properties that define only the foreground color are:

- Cell-Current-Foreground-Color
- Cell-Entry-Foreground-Color
- Cell-Foreground-Color
- Colors
- Column-Foreground-Color
- Cursor-Foreground-Color
- Divider-Color
- Drag-Foreground-Color
- End-Color
- Fill-Color
- Fill-Color2
- Foreground-Color
- Heading-Divider-Color
- Heading-Foreground-Color
- Row-Cursor-Foreground-Color
- Row-Foreground-Color
- Row-Foreground-Color-Pattern
- Sunday-Foreground
- Weekday-Foreground

Using RGB

A more flexible and precise way to define colors is RGB.

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

RGB colors can be specified for all those properties that define either a foreground or a background color, such as FOREGROUND-COLOR and BACKGROUND-COLOR. Properties that can define both, such as COLOR, must use only the attributes explained above.

To specify an RGB color the word RGB must be put in front of the value

```
FOREGROUND-COLOR 1 | Uses the attribute 1
```

```
FOREGROUND-COLOR RGB x#489919 | Uses the RGB color #489919, a light green
```

RGB color is a number that can be calculated using the following formula:

$$Red * 2^{16} + Green * 2^8 + Blue * 2^0$$

Red, *Green* and *Blue* can range from 0 to 255.

When the RGB color is applied to a background element, transparency can be defined, too:

$$Alpha * 2^{24} + Red * 2^{16} + Green * 2^8 + Blue * 2^0$$

Alpha can range from 1 (transparent) to 127 (opaque). A value of 0 means no transparency.

In order to specify the RGB color using a variable, a signed numeric data item must be used, the RGB keyword must be omitted, and the color value must be calculated as follows:

$$([Alpha * 2^{24} +] Red * 2^{16} + Green * 2^8 + Blue * 2^0) * -1$$

In practice, putting the RGB keyword before the color value or multiplying the value by -1 have the same effect.

When a color property set with RGB is queried, the INQUIRE statement returns the the decimal negative number representing the RGB color.

Note: Colors are set and queried in the same way regardless of their type. The runtime distinguishes the two kinds of colors by using positive values for attributes and negative values for RGB. Since zero is a non-negative value, you shouldn't use the value zero along with the RGB syntax, use 1 instead and you will obtain the same color.

Layout managers

isCOBOL includes a layout manager facility that can be applied to help manage some of the tricky aspects of a screen's layout. A layout manager is a specialized piece of software that is attached to a window and that manages the placement and size of controls in that window. Individual layout managers have their own rules regarding how controls are sized and placed.

By default, a window does not have a layout manager attached to it. For such windows, controls are sized and placed according to their Line, Column, Lines and Size properties. When a layout manager is attached to a window, the layout manager determines the size and placement of controls, although it is free to use the Line, Column, Lines and Size properties to help make decisions. A control can provide additional information about itself, including special size and placement parameters, to the layout manager through the Layout-Data property. The precise meaning of Layout-Data varies from layout manager to layout manager.

Layout managers operate whenever a new control is placed in the window or the window is resized.

Layout managers can be applied on Window, Tool-Bar and Ribbon through the Layout-Manager property. isCOBOL supports the following layout managers:

- [LM-RESIZE](#)
- [LM-RESPONSIVE](#)
- [LM-SCALE](#)
- [LM-WRAP](#)

The [isresize.def](#) Copybook includes the data items that you can use to set the Layout-Manager property.

LM-RESIZE

LM-RESIZE automatically resizes and moves controls when the window is resized, allowing to cover all the space available.

LM-RESIZE acts on any control that has a non-zero Layout-Data value. The exact value determines what actions the resize manager takes. The resize manager assumes that it has complete control over the size and placement of controls that have Layout-Data. After such a control has been displayed, the program should not modify it in a way that changes its size or position (that is the job of the resize manager). Doing so may result in improper resizing or repositioning by LM-RESIZE.

For LM-RESIZE, a control's Layout-Data property may be a combination of any of the following values. To combine values, simply add them together. The names of the values come from [isresize.def](#).

Constant name	Effect
rlm-resize-x	Causes the control to grow and shrink horizontally as the window changes width.
rlm-move-x	Causes the control to reposition itself horizontally as the window changes width.
rlm-no-min-x	Without this, the resize manager will not reposition or resize a control horizontally to be less than its design values. This prevents the control from disappearing or colliding with other controls if the user makes the window too small.
rlm-resize-y	Causes the control to grow and shrink vertically as the window changes width.
rlm-move-y	Causes the control to reposition itself vertically as the window changes width.
rlm-no-min-y	Without this, the resize manager will not reposition or resize a control vertically to be less than its design values. This prevents the control from disappearing or colliding with other controls if the user makes the window too small.

The following are also found in [isresize.def](#). These are not unique values, but useful combinations of the preceding values.

Constant name	Effect
rlm-resize-x-any	rlm-resize-x + rlm-no-min-x
rlm-move-x-any	rlm-move-x + rlm-no-min-x
rlm-resize-y-any	rlm-resize-y + rlm-no-min-y

Constant name	Effect
rlm-move-y-any	rlm-move-y + rlm-no-min-y
rlm-resize-both	rlm-resize-x + rlm-resize-y
rlm-resize-both-any	rlm-resize-x-any + rlm-resize-y-any
rlm-move-both	rlm-move-x + rlm-move-y
rlm-move-both-any	rlm-move-x-any + rlm-move-y-any

LM-RESPONSIVE

LM-RESPONSIVE automatically resizes, hides, shrinks, or enlarges controls, to make the screen look good on different screen widths.

The different widths are defined along with the layout manager as follows:

```
HANDLE OF LAYOUT-MANAGER LM-RESPONSIVE "breakpoints"
```

Where *breakpoints* is a series of logical names followed by a size in the format:

```
name=size [cells|pixels]
```

Where *name* is a free name, *size* is a numeric value and *cells* or *pixels* specify the measurement unit. If neither *cells* nor *pixels* is specified, then *pixels* is assumed. If *size* is decimal, the dot symbol must be used as decimal separator.

Multiple breakpoints can be defined. Their description must be separated by comma.

The following snippet defines a LM-RESPONSIVE layout manager that operates on three different screen widths: small, medium and large:

```
77 responsive-layout handle of layout-manager, lm-responsive
    "small=14 cells, medium=40 cells, large=69 cells" .
```

When the window width changes in the range between 14 and 40 cells, the small breakpoint is used.

When the window width changes in the range between 40 and 69 cells, the medium breakpoint is used.

When the window width is increased to 69 cells or more, the large breakpoint is used.

When the window width is reduced below 14 cells, no specific breakpoint is used.

When the window width changes from a breakpoint to the other, the controls are redesigned by LM-RESPONSIVE according to their Layout-Data property.

Layout-Data must be set to an alphanumeric value that is a combination of one or more of the following entries separated by space:

visible-breakpoint	The control is made visible when the window width is in the range specified by <i>breakpoint</i> (unless its Visible property is set to false)
hidden-breakpoint	The control is hidden when the window width is in the range specified by <i>breakpoint</i>

<i>line-breakpoint value</i> [cells pixels]	The control is positioned on the line number specified by <i>value</i> when the window width is in the range specified by <i>breakpoint</i>
<i>lines-breakpoint value</i> [cells pixels]	The control is resized to the height specified by <i>value</i> when the window width is in the range specified by <i>breakpoint</i>
<i>col-breakpoint value</i> [cells pixels]	The control is positioned on the column number specified by <i>value</i> when the window width is in the range specified by <i>breakpoint</i>
<i>column-breakpoint value</i> [cells pixels]	The control is positioned on the column number specified by <i>value</i> when the window width is in the range specified by <i>breakpoint</i>
<i>size-breakpoint value</i> [cells pixels]	The control is resized to the width specified by <i>value</i> when the window width is in the range specified by <i>breakpoint</i>

Where *breakpoint* is one of the breakpoints specified in the layout manager definition, *value* is a numeric value (with the dot symbol as decimal separator) and *cells* or *pixels* are the measurement unit. If neither *cells* nor *pixels* is specified, then *pixel* is assumed.

If *visible* or *hidden* is specified only for one breakpoint, the opposite is assumed for the other breakpoints. For example, given the small, medium and large breakpoints described above, we can say that:

```
Layout-Data "visible-small"
```

is equivalent to:

```
Layout-Data "visible-small hidden-medium hidden-large"
```

When the window width changes inside the range between two breakpoints, the rules of [LM-SCALE](#) are applied.

You can instruct LM-SCALE by adding one or more of the following entries to the control's Layout-Data:

<i>move-x-breakpoint</i>	The control is moved on the x-axis
<i>resize-x-breakpoint</i>	The control is resized on the x-axis
<i>no-min-x-breakpoint</i>	Allow the control to be resized or moved on the x-axis below its design values
<i>move-y-breakpoint</i>	The control is moved on the y-axis
<i>resize-y-breakpoint</i>	The control is resized on the y-axis
<i>no-min-y-breakpoint</i>	Allow the control to be resized or moved on the y-axis below its design values
<i>move-x-any-breakpoint</i>	Combination of move-x and no-min-x
<i>resize-x-any-breakpoint</i>	Combination of resize-x and no-min-x
<i>move-y-any-breakpoint</i>	Combination of move-y and no-min-y
<i>resize-y-any-breakpoint</i>	Combination of resize-y and no-min-y
<i>move-both-breakpoint</i>	Combination of move-x and move-y
<i>resize-both-breakpoint</i>	Combination of resize-x and resize-y

<i>move-both-any-breakpoint</i>	Combination of move-x, move-y, no-min-x and no-min-y
<i>resize-both-any-breakpoint</i>	Combination of resize-x, resize-y, no-min-x and no-min-y
<i>no-scale-breakpoint</i>	No action

Where *breakpoint* is one of the breakpoints specified in the layout manager definition. If "-breakpoint" is omitted, then the entry is applied to all the breakpoints defined in the layout manager.

The following snippet shows how to set Layout-Data to make a entry-field go on the next line, column 2 with no scale action when the screen is reduced from the "medium" to "small":

```
03 ef-1 entry-field
   line 2, col 10, size 10 cells
   layout-data "line-small 3 col-small 2 no-scale"
```

LM-SCALE

LM-SCALE moves and resizes controls proportionally to the new dimensions of the window.

For each resize action:

$\text{deltaX} = \text{current window size} / \text{previous window size}$,

$\text{deltaY} = \text{current window lines} / \text{previous window lines}$.

For each control:

RESIZE-X	$\text{size} = \text{previous size} * \text{deltaX}$.
RESIZE-Y	$\text{lines} = \text{previous lines} * \text{deltaY}$.
MOVE-X	$\text{column} = \text{previous column} * \text{deltaX}$.
MOVE-Y	$\text{line} = \text{previous line} * \text{deltaY}$.

If the Layout-Data property is not set, each control has its own default behavior.

The list below describes the default behavior of each control:

BAR	MOVE-BOTH-ANY
BITMAP	MOVE-BOTH-ANY
CHECK-BOX	MOVE-BOTH-ANY
COMBO-BOX	MOVE-BOTH-ANY + RESIZE-X-ANY
DATE-ENTRY	MOVE-BOTH-ANY
ENTRY-FIELD	MOVE-BOTH-ANY + RESIZE-X-ANY
ENTRY-FIELD Multiline	MOVE-BOTH-ANY + RESIZE-BOTH-ANY
FRAME	MOVE-BOTH-ANY + RESIZE-BOTH-ANY

GRID	MOVE-BOTH-ANY + RESIZE-BOTH-ANY
JAVA-BEAN	MOVE-BOTH-ANY
LABEL	MOVE-BOTH-ANY
LIST-BOX	MOVE-BOTH-ANY + RESIZE-BOTH-ANY
PUSH-BUTTON	MOVE-BOTH-ANY
RADIO-BUTTON	MOVE-BOTH-ANY
SCROLL-BAR HORIZONTAL	MOVE-BOTH-ANY + RESIZE-X-ANY
SCROLL-BAR VERTICAL	MOVE-BOTH-ANY + RESIZE-Y-ANY
SLIDER HORIZONTAL	MOVE-BOTH-ANY + RESIZE-X-ANY
SLIDER VERTICAL	MOVE-BOTH-ANY + RESIZE-Y-ANY
TAB CONTROL	MOVE-BOTH-ANY + RESIZE-BOTH-ANY
TREE-VIEW	MOVE-BOTH-ANY + RESIZE-BOTH-ANY
WEB-BROWSER	MOVE-BOTH-ANY + RESIZE-BOTH-ANY

The user can also specify custom values for some controls directly in the handle definition, without using the Layout-Data property.

Example:

```
77 h-layout handle of layout-manager, lm-scale "configuration string".
```

A sample configuration string is provided in the [isresize.def](#) Copybook.

The configuration string is composed of a series of <control-type>=<value> separated by a spaces or comma. <control-type>=<value> must be a unique word. No space may be put inside. The string is case insensitive.

Example:

```
77 h-layout handle of layout-manager lm-scale "bar=119 textarea=119 label=119
frame=119".
```

The list below describes the <control-type> you can use in the configuration string.

BAR	bar
BITMAP	bitmap
CHECK-BOX	checkbox
COMBO-BOX	combobox
DATE-ENTRY	dateentry
ENTRY-FIELD	entryfield

ENTRY-FIELD Multiline	textarea
ENTRY-FIELD SPINNER	spinner
FRAME	frame
GRID	grid
JAVA-BEAN	javabean
LABEL	label
LIST-BOX	listbox
PUSH-BUTTON	pushbutton
RADIO-BUTTON	radiobutton
HORIZONTAL SCROLLBAR	hscrollbar
VERTICAL SCROLLBAR	vscrollbar
HORIZONTAL SLIDER	hslider
VERTICAL SLIDER	vslider
TAB CONTROL	tab
TREE-VIEW	treeview
WEB-BROWSER	webbrowser

<value> is a numeric value that is the sum of the constant values that must be used for the control.

In the above sample, 119 is the sum between rlm-resize-x, rlm-move-x, rlm-no-min-x, rlm-resize-y, rlm-move-y, rlm-no-min-y.

Note: inquiring controls Lines and Size properties after the window has been resized returns the initial values and not the new values when the Layout-Manager is LM-SCALE.

LM-WRAP

LM-WRAP automatically redistribute controls on multiple lines when the window is resized, allowing to cover all the space available.

This layout manager is supported only on the Ribbon control.

Help automation

Help automation support is based on the concept of a help ID. A help ID is a special integer value assigned to a control. When a help request is sent to the help processor, the help ID of the associated control is sent as a parameter. Typically, each control is assigned a unique value. This allows the help processor to uniquely respond to each control. To create help that responds to the window, rather than an individual control within it, you can give all of the controls within a window the same help ID. Or you can mix the two approaches by giving some individual controls unique help IDs, while the remaining controls get a shared help ID. Because help IDs are associated with controls, help automation can't be used with character-based ACCEPT fields.

Whether the control is defined in the Screen Section or in a DISPLAY statement, help IDs are assigned with the HELP-ID phrase. You can easily assign a screen-wide help ID to a window by specifying a HELP-ID for the top-level group item in the Screen Section description. You can override the screen-wide ID for a specific control by including the HELP-ID phrase in that control's definition. After setting up the help IDs, you must assign the help mode an exception value. This is done with the Format 13 SET statement. For example, in order to have the F1 key trigger the item help, use

```
SET EXCEPTION VALUES 1 TO ITEM-HELP
```

After the exception values are assigned, any control, menu item, or key that produces the specified exception value will produce the associated help action. The last step in setting up help automation is to define the name of the help processor program. The help processor's entry point is always a COBOL program. The program can be the help processor itself. The program can also be a bridge to some other help processor, for example, on Windows, you may interface the Windows Help through the \$WINHELP library routine.

The help processor is named by the value of the configuration property iscobol.help_program. If such property is undefined, no help processor is called.

The help processor is passed only one parameter, the EVENT STATUS data item. It contains the CMD-HELP event that generated the CALL. The CMD-HELP event contains all of the information needed to process the help request:

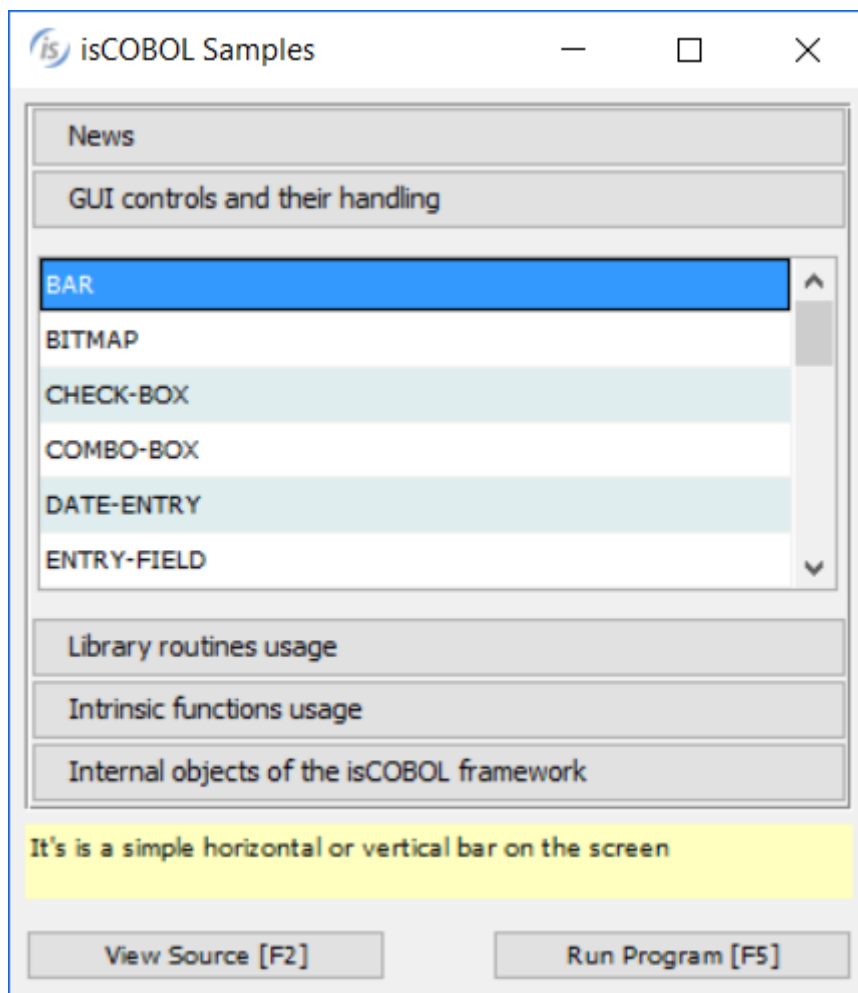
- the control's handle (in EVENT-CONTROL-HANDLE),
- the control's ID (in EVENT-CONTROL-ID),
- the control's help ID (in EVENT-DATA-2) and
- the handle of the control's owning window (in EVENT-WINDOW-HANDLE).

Chapter 2

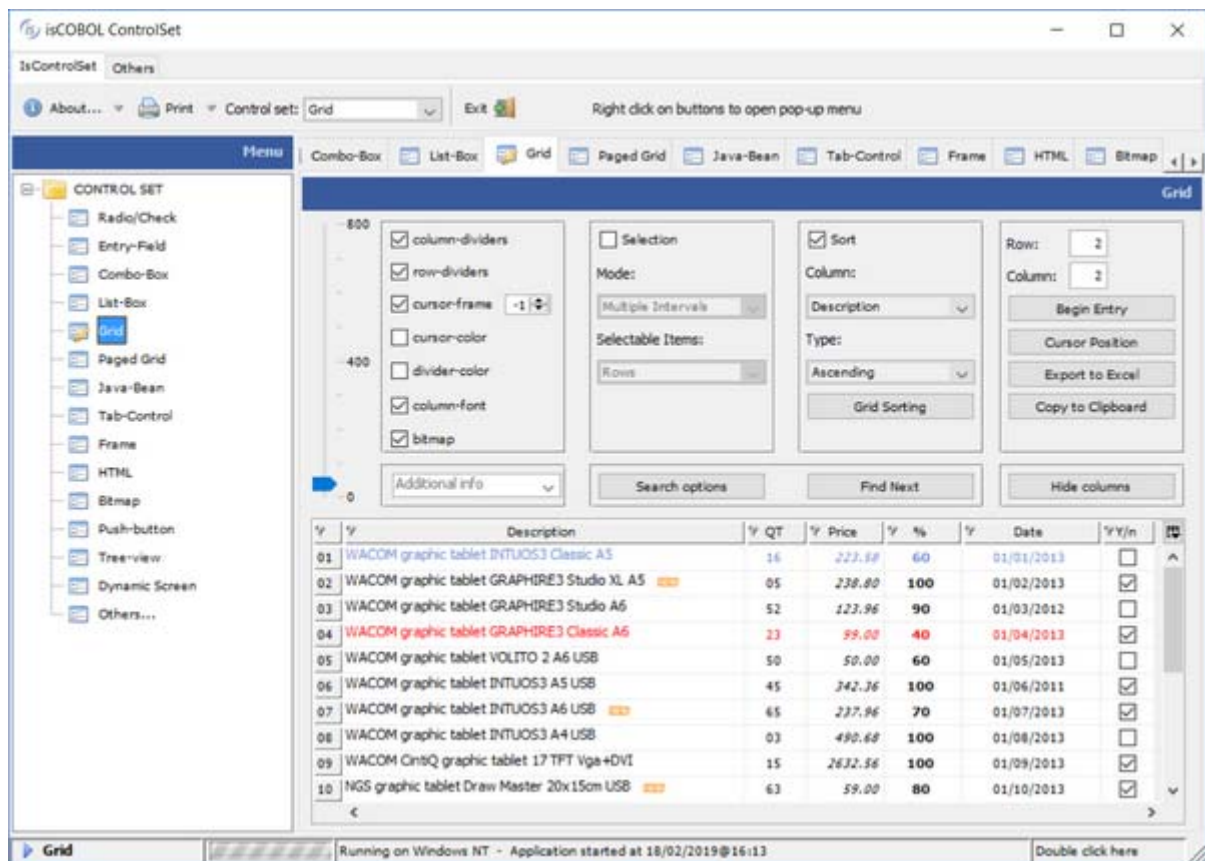
Controls Reference

This chapter describes all the graphical controls that can appear in the program Screen Section.

Basic sample programs for each control are available amongst the isCOBOL Samples.



The advanced usage of graphical controls is shown in the isCOBOL Demo.



BAR



A Bar is a line consisting of a number of one-pixel rows. Each row can be rendered with different color, size or shading, in order to match all the programmer's needs.

When the [Lines](#) property is zero, the bar is horizontal. When the [Size](#) property is zero, the bar is vertical.

Properties

The following properties are applicable to the BAR control: [Col](#), [Color](#), [Colors](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Id](#), [Layout-data](#), [Leading-Shift](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Position-Shift](#), [Shading](#), [Size](#), [Trailing-Shift](#), [Visible](#), [Width](#).

[Col | Column | Pos | Position]

This property allows you to specify the [BAR](#) control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Bar control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name

and its value. The horizontal position of the Bar control will be relative to the ending position of the prior Screen Section item.

When the Bar control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Bar, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a bar at column 5.0 on the screen section definition

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Bar control. The Bar control has only a Foreground color and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a bar on screen section with color 7

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Colors

The Bar control consists of a number of rows defined with the [Width](#) property that make the Bar control thicker or thinner.

The color of each single row is defined by this property. If not set, each row will be drawn with the color set to the [Color](#) or [Foreground-Color](#) properties.

Since this property can be set for each row, a list of values is needed in order to determine the color of each row.

When values are enclosed between parentheses, a new list is defined at once. The snippet below defines a

gray fading Bar control.

```
WIDTH    = 8
COLORS   = (RGB x#000000,
            RGB x#222222,
            RGB x#444444,
            RGB x#666666,
            RGB x#888888,
            RGB x#AAAAAA,
            RGB x#CCCCCC,
            RGB x#EEEEEE)
```

See ["Color management"](#) for further details.

Example - Define a bar on screen section with 3 different colors for a width 3

```
screen section.
...
03 screen-1-br-1 Bar
   line 8.0
   column 5.0
   color 7
   size 45.0 cells
   id 2
   width = 3
   colors =
      (rgb x#222222,
       rgb x#444444,
       rgb x#cccccc)
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a bar on screen section with a css-base-style-name

```
screen section.
...
03 screen-1-br-4 Bar
   line 5.5
   column 2.7
   size 9.6 cells
   id 4
   css-base-style-name "css-bars"
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a bar on screen section with a css-style-name

```
screen section.  
...  
03 screen-1-br-4 Bar  
    line 5.5  
    column 2.7  
    size 9.6 cells  
    id 4  
    css-style-name "css-bars"  
    .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a bar

```
procedure division.  
...  
    modify screen-1-br-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property is not considered by the Bar control.

Font

This property specifies the font that may be used to compute the height and the width of the Bar control. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a bar using an Arial font handle as the font reference for sizing it

```
working-storage section.
copy "isfonts.def".
77 Arial-0v0 handle of font.
...

screen section.
...
03 screen-1-br-2 Bar
   line 7.6
   column 4.9
   size 49.4 cells
   font Arial-0v0
   id 2
   shading(-2,0,0,2,2)
   width 5
   .
...
procedure division.
...
is-load-fonts.
   initialize wfont-data arial-0v0.
   move 0 to wfont-size.
   move "Arial" to wfont-name.
   set wfont-bold to false.
   set wfont-italic to false.
   set wfont-underline to false.
   set wfont-strikeout to false.
   set wfont-fixed-pitch to false.
   call "w$font" using wfont-get-font arial-0v0 wfont-data.
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Bar control. See ["Color management"](#) for further details.

Example - Define a bar on screen section with a foreground-color in rgb

```
screen section.
...
03 screen-1-br-1 Bar
   line 5.5
   column 5.1
   size 46.8 cells
   foreground-color rgb 32960
   id 1
   .
```

Help-Id

This property is not considered by the Bar control.

Id

This property allows you to assign a unique ID to the Bar control.

Example - Define a bar on screen section with ID 3

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.5  
   column 5.1  
   size 46.8 cells  
   id 3  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a vertical bar, that allows X and Y resizing

```
screen section.  
...  
03 screen-1-br-3 Bar  
   line 10.0  
   column 5.0  
   lines 22.0 cells  
   id 3  
   layout-data 17 |*> To allow X and Y resizing  
   .
```

Leading-Shift

The Bar control consists of a number of rows defined with the [Width](#) property that make the Bar control thicker or thinner.

The width of each single row can be adjusted. Sometimes it is needed to obtain a better look where the left end (or the top, when the Bar control is vertical) of a Bar control connects to another one.

The value set here is the amount of pixels that are subtracted from the row. Use negative values to add pixels.

Since this property can be set for each row, a list of values is needed in order to determine the exact size of each row.

When values are enclosed between parentheses, a new list is defined at once. The snippet below defines a Bar control that is 5 pixels wide and begins with a 45 degree angle.

```
WIDTH          = 5  
LEADING-SHIFT = (0, 1, 2, 3, 4)
```

Setting this property to a value greater than or equal to 999 resets the list.

When a single value greater than zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Example - Define a user-defined appearance

```
modify screen-1-br-1, leading-shift = 999 | resets the list of values
perform varying rowidx from 1 by 1 until rowidx > rowcount
    modify screen-1-br-1, leading-
shift= customshift(rowidx) | sets the size of the next row.
end-perform
```

Line

This property allows you to specify the Bar control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Bar control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Bar control will be relative to the starting position of the prior Screen Section item.

When the Bar control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).
03 Bar, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a bar at line 8.0 on the screen section definition

```
screen section.
...
03 screen-1-br-1 Bar
    line 8.0
    column 5.0
    color 7
    size 45.0 cells
    id 2
.
```

Lines

This property allows you to specify the height of the Bar control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Bar control is still computed in CELLS, but the cell size is based on the font set for the Bar control with the [Font](#) property. If no font has been defined for the Bar control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a vertical bar on the screen section definition

```
screen section.  
...  
03 screen-1-br-3 Bar  
   line 10.0  
   column 5.0  
   lines 22.0 cells  
   id 3  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a vertical bar, that allows X and Y resizing and will have a MAX-HEIGHT that limits how large it could be when resizing the window

```
screen section.  
...  
03 screen-1-br-3 Bar  
   line 10.0  
   column 5.0  
   lines 22.0 cells  
   id 3  
   max-height 44.0  
   layout-data 17 |*> To allow X and Y resizing  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define an horizontal bar, that allows X resize only and will have a MAX-WIDTH that limits how wide it could be when resizing the window

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.0  
   column 5.0  
   size 46.0 cells  
   id 1  
   max-width 90.0  
   layout-data 1 |*> To allow X resizing only  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a vertical bar, that allows X and Y resizing and will have a MIN-HEIGHT that limits how short it could be when resizing the window

```
screen section.  
...  
03 screen-1-br-3 Bar  
   line 10.0  
   column 5.0  
   lines 22.0 cells  
   id 3  
   min-height 11.0  
   layout-data 17 |*> To allow X and Y resizing  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define an horizontal bar, that allows X resize only and will have a MIN-WIDTH that limits how narrow it could be when resizing the window

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.0  
   column 5.0  
   size 46.0 cells  
   id 1  
   min-width 23.0  
   layout-data 1 |*> To allow X resizing only  
   .
```

Pop-Up Menu

This property is not considered by the Bar control.

Position-Shift

With this property it is possible to adjust the vertical position of the horizontal bars or the horizontal position of the vertical bars. The adjustment is in pixels. Negative values can be used to shift the bar up or left.

Example - Define 2 horizontal bars in the same line, but define the 2nd one to shift 5 pixels, the visible result

will be that the second one will show 5 pixels below the 1st one although they are defined on the same line

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.0  
   column 2.0  
   size 9.0 cells  
   id 1  
   .  
03 screen-1-br-2 Bar  
   line 5.0  
   column 12.0  
   size 46.0 cells  
   id 2  
   position-shift 5  
   .
```

Shading

The Bar control consists of a number of rows defined with the [Width](#) property that make the Bar control thicker or thinner.

The color of each single row is defined by the [Colors](#) property. If not set, each row will be drawn with the color set in the [Color](#) or [Foreground-Color](#) properties.

To obtain a 3-D effect, some rows must be drawn with a lighter or a darker color. This property allows you to draw lighter or darker rows, changing their brightness instead of their color definition

Valid values are:

-2	Very dark, usually black.
-1	Darker than normal.
0	Unchanged.
1	Lighter than normal.
2	Very light, usually white.

Since this property can be set for each row, a list of values is needed in order to determine the brightness of each row.

When values are enclosed between parentheses, a new list is defined at once.

Example - A typical 3-D Bar control.

```
screen section.  
...  
03 screen-1-br-2 Bar  
   line 5.0  
   column 12.0  
   size 46.0 cells  
   id 2  
   color white low  
   width 2  
   shading (-1, 1)  
   .
```

Size

This property allows you to specify the size of the Bar control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Bar control is still computed in CELLS, but the cell size is based on the font set for the Bar control with the [Font](#) property. If no font has been defined for the Bar control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a horizontal bar on screen section with size 46.0 cells

```
screen section.  
...  
03 screen-1-br-2 Bar  
   line 5.0  
   column 12.0  
   size 46.0 cells  
   id 2  
   .
```

Trailing-Shift

The Bar control consists of a number of rows defined with the [Width](#) property that make the Bar control thicker or thinner.

The width of each single row can be adjusted. Sometimes it is needed to obtain a better look where the right end (or the bottom, when the Bar control is vertical) of a Bar control connects to another one.

The value set here is the amount of pixels that are added to the row. Use negative values to subtract pixels.

Since this property can be set for each row, a list of values is needed in order to determine the exact size of each row.

When values are enclosed between parentheses, a new list is defined at once. The snippet below defines a Bar control that is 5 pixels wide and ends with a 45 degree angle.

```
WIDTH          = 5  
TRAILING-SHIFT = (0, 1, 2, 3, 4)
```

Setting this property to a value greater than or equal to 999 resets the list.

When a single value greater than zero is set, it is appended to the list. This is useful to define a user-defined appearance

Example - Define a user-defined appearance.

```
modify screen-1-br-2, trailing-shift = 999 | resets the list of values
perform varying rowidx from 1 by 1 until rowidx > rowcount
  modify screen-1-br-2, trailing-
shift = customshift(rowidx) | sets the size of the next row.
end-perform
```

Visible

This property assumes a value of "0" if the Bar control is not visible, "1" if it is visible.

Example - Make a bar invisible and visible during runtime

```
procedure division.
...
  if bars-invisible |*> Any 88 condition name
    modify screen-1-br-2 visible 0 |*> Making it invisible
  else
    modify screen-1-br-2 visible 1 |*> Making it visible
  end-if
```

Width

This property, expressed in pixels, sets the thickness of the bar.

Example - Define a bar on screen section with WIDTH 3

```
screen section.
...
03 screen-1-br-1 Bar
  line 8.0
  column 5.0
  color 7
  size 45.0 cells
  id 2
  width = 3
.
```

Styles

The following styles are applicable to the BAR control: **Bold**, **Dashed**, **Dot-Dash**, **Dotted**, **Height-In-Cells**, **High**, **Highlight**, **Low**, **Lowlight**, **Permanent**, **Standard**, **Temporary**, **Width-In-Cells**.

{ Dotted | Dashed | Dot-Dash }

Dotted	A dotted line is displayed
Dashed	A dashed line is displayed
Dot-Dash	A line with alternate dots and dashes is displayed

Example - Define 3 bars using these 3 alternative styles

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.0  
   column 2.0  
   size 9.0 cells  
   id 1  
   dotted  
   .  
  
03 screen-1-br-2 Bar  
   line 7.0  
   column 2.0  
   size 9.0 cells  
   id 2  
   dashed  
   .  
  
03 screen-1-br-3 Bar  
   line 9.0  
   column 2.0  
   size 9.0 cells  
   id 3  
   dot-dash  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Bar control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a vertical bar on the screen section definition expressing height in cells

```
screen section.  
...  
03 screen-1-br-3 Bar  
   line 10.0  
   column 5.0  
   lines 22.0  
   id 3  
   height-in-cells  
   .
```


{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define 3 bars with different foreground intensity, one BOLD, one LOW and one STANDARD

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.0  
   column 2.0  
   size 9.0 cells  
   id 1  
   bold  
   .  
  
03 screen-1-br-2 Bar  
   line 7.0  
   column 2.0  
   size 9.0 cells  
   id 2  
   low  
   .  
  
03 screen-1-br-3 Bar  
   line 9.0  
   column 2.0  
   size 9.0 cells  
   id 3  
   standard  
   .
```

{ **Permanent** | **Temporary** }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define 2 bars on screen section, the first permanent (by default) and the other temporary

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 5.0  
   column 2.0  
   size 9.0 cells  
   id 1  
   .  
  
03 screen-1-br-2 Bar  
   line 7.0  
   column 2.0  
   size 9.0 cells  
   id 2  
   temporary  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Bar control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define an horizontal bar on screen section with size 46.0 expressed in cells

```
screen section.  
...  
03 screen-1-br-2 Bar  
   line 5.0  
   column 12.0  
   size 46.0  
   id 2  
   width-in-cells  
   .
```

Events

The Bar control doesn't fire events.

BITMAP



A Bitmap is a control that shows a previously loaded image. Native and emulated transparency is supported. Emulated animation is provided using a bitmap strip and setting the [Bitmap-Start](#), [Bitmap-End](#) and [Bitmap-Timer](#) properties. A bitmap strip is a series of images of equal width that are strung together horizontally in a single bitmap file.

Properties

The following properties are applicable to the BITMAP control: [Background-Color](#), [Bitmap-End](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Scale](#), [Bitmap-Start](#), [Bitmap-Timer](#), [Col](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Size](#), [Transparent-Color](#), [Visible](#).

Background-Color

This property is not considered by the Bitmap control.

Bitmap-End

This property is used to define the last image in a bitmap strip to be used for a bitmap animation.

Example - Define a bitmap control that will show image strips from 3 (bitmap-start) to 6 (bitmap-end) and each image will be shown 50 hundredths (bitmap-timer) of a second

```
working-storage section.  
...  
77 icon-png pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-bi-1 Bitmap  
   line 14.9  
   column 50.5  
   size 17 pixels  
   lines 36 pixels  
   id 5  
   bitmap-handle icon-png  
   bitmap-number 3  
   bitmap-start 3  
   bitmap-end 6  
   bitmap-timer 50  
   .
```

Bitmap-Handle

This property identifies the bitmap handle to be used. To obtain a bitmap handle use the W\$BITMAP library routine with the WBITMAP-LOAD op-code.

If this property points to an animated GIF, then the Bitmap control renders the animation.

Example - Define a bitmap control with a handle that is set in procedure division

```
working-storage section.
...
77 jlogo-jpg pic s9(9) comp-4.
...
screen section.
...
03 screen-1-bi-1 Bitmap
   line 10.0
   column 48.0
   size 130 pixels
   lines 150 pixels
   id 5
   bitmap-scale 1
   bitmap-handle jlogo-jpg
   bitmap-number 1
   .
...
procedure division.
...
is-load-bitmaps.
   call "w$bitmap"
       using wbitmap-load "jlogo.jpg"
       giving jlogo-jpg.
```

Bitmap-Number

This property identifies the image to be displayed when the Bitmap control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a bitmap control and use the second image on a bitmap strips that contains several images

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-bi-1 Bitmap
   line 14.0
   column 50.0
   size 17 pixels
   lines 30 pixels
   id 5
   bitmap-handle icon-png
   bitmap-number 2
   .
```

Bitmap-Scale

This property specifies what to do if the image dimensions don't fit the Bitmap area identified by [Lines](#) and [Size](#) properties.

The possible values for this property are:

0	The image is not altered. In this case, if the image is too large, it will be truncated, if it's too small, it will be aligned to the top left corner of the Bitmap area. This is also the default behavior when Bitmap-Scale is not set.
1	The image is resized to fit completely the Bitmap area. The aspect ratio may be altered.
2	The image is resized maintaining the aspect ratio. The resized image may not fit completely the Bitmap area.

If the Bitmap control is resized by a Layout-Manager and Bitmap-Scale is set to "1" or "2", then the image is resized along with the window.

Example - Define a bitmap control that will display a scaled image

```
working-storage section.  
...  
77 jlogo-jpg pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-bi-1 Bitmap  
   line 10.0  
   column 48.0  
   size 130 pixels  
   lines 150 pixels  
   id 5  
   bitmap-scale 1  
   bitmap-handle jlogo-jpg  
   bitmap-number 1  
   .
```

Bitmap-Start

This property is used to define the first image in a bitmap strip to be used for a bitmap animation.

Example - Define a bitmap control that will show image strips from 3 (bitmap-start) to 6 (bitmap-end) and

each image will be shown 50 hundredths (bitmap-timer) of a second

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-bi-1 Bitmap
   line 14.9
   column 50.5
   size 17 pixels
   lines 36 pixels
   id 5
   bitmap-handle icon-png
   bitmap-number 3
   bitmap-start 3
   bitmap-end 6
   bitmap-timer 50
.
```

Bitmap-Timer

This property sets the amount of time any bitmap is displayed (for animation) from Bitmap-Start to Bitmap-End and is expressed in hundredths of seconds.

Example - Define a bitmap control that will show image strips from 3 (bitmap-start) to 6 (bitmap-end) and each image will be shown 50 hundredths (bitmap-timer) of a second

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-bi-1 Bitmap
   line 14.9
   column 50.5
   size 17 pixels
   lines 36 pixels
   id 5
   bitmap-handle icon-png
   bitmap-number 3
   bitmap-start 3
   bitmap-end 6
   bitmap-timer 50
.
```

[Col | Column | Pos | Position]

This property allows you to specify the Bitmap control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Bitmap control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Bitmap control will be relative to the ending position of the

prior Screen Section item.

When the Bitmap control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).
03 Bitmap, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a bitmap at column 5.0 on the screen section definition

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-bi-1 Bitmap
   line 8.0
   column 5.0
   size 17 pixels
   lines 36 pixels
   id 5
   bitmap-handle icon-png
   id 2
   .
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a bitmap control with css base style (valid for EIS WD2 deployment)

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-bi-1 Bitmap
   line 14.0
   column 50.0
   size 17 pixels
   lines 30 pixels
   id 5
   css-base-style-name "bitmap-css-style"
   bitmap-handle icon-png
   bitmap-number 2
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a bitmap control with css style (valid for EIS WD2 deployment)

```
working-storage section.  
...  
77 icon-png pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-bi-1 Bitmap  
    line 14.0  
    column 50.0  
    size 17 pixels  
    lines 30 pixels  
    id 5  
    css-style-name "bitmap-css-style"  
    bitmap-handle icon-png  
    bitmap-number 2  
    .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a bitmap

```
procedure division.  
...  
    modify screen-1-bi-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Bitmap control is disabled, "1" if it is enabled.

Example - Define a bitmap that is initially disabled, enabled it later in procedure division

```
screen section.  
...  
  03 screen-1-bi-2 Bitmap  
    line 23.3  
    column 51.4  
    size 83 pixels  
    lines 97 pixels  
    enabled 0  
    id 8  
    .  
...  
procedure division.  
...  
  modify screen-1-bi-2 enabled 1.  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a bitmap including event handling for mouse-enter and mouse-exit only

```
screen section.  
...  
  03 screen-1-bi-3 Bitmap  
    line 15.8  
    column 33.5  
    size 63 pixels  
    lines 68 pixels  
    id 9  
    event-list ( msg-mouse-enter msg-mouse-exit)  
    exclude-event-list 0  
    .  

```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a bitmap excluding event handling for mouse-enter and mouse-exit

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.8  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   event-list ( msg-mouse-enter msg-mouse-exit)  
   exclude-event-list 1  
   .
```

Font

This property specifies the font that may be used to compute the height and the width of the Bitmap control. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a bitmap using an Arial font handle as the font reference for sizing it

```
working-storage section.  
copy "isfonts.def".  
77 Arial-0v0    handle of font.  
77 jlogo-jpg0  pic s9(9) comp-4.  
...  
  
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.8  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   font Arial-0v00  
   id 9  
   bitmap-scale 1  
   bitmap-handle jlogo-jpg0  
   bitmap-number 1  
   .  
...  
  
procedure division.  
...  
is-load-fonts.  
   initialize wfont-data arial-0v0.  
   move 0 to wfont-size.  
   move "Arial" to wfont-name.  
   set wfont-bold to false.  
   set wfont-italic to false.  
   set wfont-underline to false.  
   set wfont-strikeout to false.  
   set wfont-fixed-pitch to false.  
   call "w$font" using wfont-get-font arial-0v0 wfont-data.  
...  
...
```

Foreground-Color

This property is not considered by the Bitmap control.

Help-Id

This property is not considered by the Bitmap control.

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Bitmap control.

Example - Define a bitmap that shows a hint message when mouse hovers it

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.8  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   font Arial-0v00  
   id 9  
   hint "We are the right company for you"  
   .
```

Id

This property allows you to assign a unique ID to the Bitmap control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a bitmap with ID 9

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.8  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a bitmap with layout-data to resize in X and Y by the layout manager

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.8  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   layout-data 17  
   .
```

Line

This property allows you to specify the Bitmap control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Bitmap control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Bitmap control will be relative to the starting position of the prior Screen Section item.

When the Bitmap control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Bitmap, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Define a bitmap on line 15.0

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.0  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   layout-data 17  
   .
```

Lines

This property allows you to specify the height of the Bitmap control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Bitmap control is still computed in CELLS, but the cell size is based on the font set for the Bitmap control with the [Font](#) property. If no font has been defined for the Bitmap control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

By default, Bitmap lines are measured in pixels.

Example - Define a bitmap whose height is 68 pixels

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.8  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a bitmap with a maximum height in case of resizing

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.0  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   max-height 20.0  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a bitmap with a maximum width in case of resizing

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   font Arial-0v00  
   id 9  
   max-width 20.8  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a bitmap with a minimum height in case of resizing

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.0  
   size 63 pixels  
   lines 68 pixels  
   id 9  
   min-height 20.0  
.
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a bitmap with a minimum width in case of resizing

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.5  
   size 63 pixels  
   lines 68 pixels  
   font Arial-0v00  
   id 9  
   min-width 20.8  
.
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Bitmap control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a bitmap with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
    03 screen-1-bi-1 Bitmap  
       pop-up menu hmenu  
       line 15.0  
       column 33.5  
       size 63 pixels  
       lines 68 pixels  
       font Arial-0v00  
       id 9  
       .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Bitmap control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Bitmap control is still computed in CELLS, but the cell size is based on the font set for the Bitmap control with the [Font](#) property. If no font has been defined for the Bitmap control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

By default, Bitmap size is measured in pixels.

Example - Define a bitmap with a specific size

```
screen section.  
...  
    03 screen-1-bi-3 Bitmap  
       line 15.0  
       column 33.0  
       size 63 pixels  
       lines 68 pixels  
       id 9  
       .  
...
```

Transparent-Color

isCOBOL automatically recognizes transparent colors or alpha channel in images that store such information (.gif, .png). For this reason this property should be used when it is necessary to have transparency with images stored in a format that does not contain transparency information. In this case, the RGB color set here is interpreted as "transparent" and all the pixels of the image containing this color are not displayed. If this property is used with images stored in gif or png format, the transparency activated by the property is added to the transparency already present in the image file.

The RGB color value is computed according to the following formula:

```
(RED * 65536) + (GREEN * 256) + BLUE
```

Example - Define a bitmap in screen section that uses an image having yellow brackground (not transparent background) and make that background to be transparent

```
working-storage section.
...
77 img-bmp          pic s9(9) comp-4.
77 yellow-color     pic 9(9).
...
screen section.
...
03 bitmap
   line 2.0, col 2.0
   lines 128.0, size 128.0
   transparent-color yellow-color
   bitmap-handle img-bmp
   bitmap-number 1
   .
...
procedure division.
...
    |the yellow RGB is #FFFF00 (255, 255, 0)
    compute yellow-color = (255 * 65536) + (255 * 256) + 0.
    call "w$bitmap" using wbitmap-load, "img.bmp"
        giving img-bmp.
```

Visible

This property assumes a value of "0" if the Bitmap control is not visible, "1" if it is visible.

Example - Define a bitmap invisible and make it visible later on procedure division

```
screen section.
...
03 screen-1-bi-3 Bitmap
   line 15.0
   column 33.0
   size 63 pixels
   lines 68 pixels
   id 9
   visible 0
   .
...
procedure division.
...
    modify screen-1-bi-3 visible 1.
...
```


Styles

The following styles are applicable to the BITMAP control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Low](#), [Lowlight](#), [Permanent](#), [Standard](#), [Temporary](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

This style is not considered by Bitmap control.

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Bitmap control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a bitmap with its height in cells using this style

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.0  
   size 63  
   lines 68  
   height-in-cells  
   width-in-cells  
   id 9  
   visible 0  
   .
```

{ [Bold | High | Highlight] | [Low | Lowlight] | Standard }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

This style is not considered by Bitmap control.

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a bitmap with temporary style

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.0  
   size 63  
   lines 68  
   width-in-cells  
   id 9  
   temporary  
.
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Bitmap control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a bitmap with its width in cells using this style

```
screen section.  
...  
03 screen-1-bi-3 Bitmap  
   line 15.0  
   column 33.0  
   size 63  
   lines 68  
   height-in-cells  
   width-in-cells  
   id 9  
   visible 0  
.
```

Events

The following events are applicable to the BITMAP control: [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-MOUSE-CLICKED](#), [MSG-MOUSE-DBLCLICK](#), [MSG-MOUSE-ENTER](#), [MSG-MOUSE-EXIT](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions that occurred in the [MSG-INIT-MENU](#) event here.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-MOUSE-CLICKED

This event is fired when the user clicks the left button of the mouse when the mouse pointer is on a Bitmap control.

MSG-MOUSE-DBLCLICK

This event is fired when the user double-clicks the left button of the mouse when the mouse pointer is on a Bitmap control.

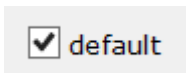
MSG-MOUSE-ENTER

This event is fired when the mouse pointer is moved on a Bitmap control.

MSG-MOUSE-EXIT

This event is fired when the mouse pointer is moved out from a Bitmap control.

CHECK-BOX



A Check-Box is used to represent a boolean value. It consists of a graphic box that shows the state of the value (on or off) and an optional text. The appearance of the graphic box, as well as the relative position of the text, can be customized.

Properties

The following properties are applicable to the CHECK-BOX control: [Background-Color](#), [Bitmap-Default](#), [Bitmap-Disabled](#), [Bitmap-Disabled-Selected](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Pressed](#), [Bitmap-Rollover](#), [Bitmap-Rollover-Selected](#), [Bitmap-Selected](#), [Bitmap-Width](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#),

Css-Style-Name, Custom-Data, Enabled, Event-List, Exception-Value, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Left-Text-Alignment, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Termination-Value, Title, Title-Position, Value, Visible.

Background-Color

This property allows you to set or retrieve the background color of the Check-Box control. See "[Color management](#)" for further details.

Example - Define a check-box in screen section with a background color

```
screen section.
...
03 screen-1-cb-1 Check-Box
   line 22.0
   column 10.0
   size 11.0 cells
   lines 3.0 cells
   background-color 3
   id 10
   title "Sugar with the coffee?"
   .
```

Bitmap-Default

This property identifies the image to be displayed when the Check-Box control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

This property conflicts with [Bitmap-Number](#). If Bitmap-Default and Bitmap-Number are used together, then the first one found in the control description is considered.

Example - Define a checkbox in screen section and define which bitmap to use when unselected from a image containing many bitmaps

```
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 116
   lines 31
   id 11
   flat
   title "My Check Box"
   bitmap-handle my-cb-png
   bitmap-default 1
   bitmap-width 20
   title-position 2
   bitmap-selected 2
   .
```

Bitmap-Disabled

This property identifies the image to be displayed when the Check-Box control is disabled. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define what bitmap to use when the check box is disabled

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "My Check Box"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-disabled 3  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   bitmap-disabled-selected 4  
   .
```

Bitmap-Disabled-Selected

This property identifies the image to be displayed when the Check-Box control is disabled and selected. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define what bitmap to use when the check box is selected but disabled

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "My Check Box"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-disabled 3  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   bitmap-disabled-selected 4  
   .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used. The [Bitmap](#) style must be set.

Example - Define a checkbox in screen section with its bitmap handle

```
working-storage section.
...
77 my-cb-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 116
   lines 31
   id 11
   flat
   title "My Check Box"
   bitmap-handle my-cb-png
   bitmap-number 1
   bitmap-width 20
   title-position 2
   bitmap-selected 2
   .
...
procedure division.
...
   call "w$bitmap" using
       wbitmap-load "my-cb.png" giving my-cb-png.
```

Bitmap-Number

This property identifies the image to be displayed when the Check-Box control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define which bitmap to use when unselected from an

image containing many bitmaps

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "E-mail results?"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   .
```

Bitmap-Pressed

This property identifies the image to be displayed when the Check-Box control is checked. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define which bitmap to use when it gets pressed from an image containing many bitmaps

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "E-mail results?"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-pressed 7  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   .
```

Bitmap-Rollover

This property identifies the image to be displayed when the mouse pointer is moved over an unchecked Check-Box control. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define which bitmap to use when the mouser rolls over it

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "E-mail results?"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-disabled 3  
   bitmap-rollover 5  
   bitmap-width 20  
   title-position 2  
   bitmap-rollover-selected 6  
   bitmap-selected 2  
   bitmap-disabled-selected 4  
   .
```

Bitmap-Rollover-Selected

This property identifies the image to be displayed when the mouse pointer is moved over a checked Check-Box control. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define which bitmap to use when the mouser rolls over it and it is selected

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "E-mail results?"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-disabled 3  
   bitmap-rollover 5  
   bitmap-width 20  
   title-position 2  
   bitmap-rollover-selected 6  
   bitmap-selected 2  
   bitmap-disabled-selected 4  
   .
```

Bitmap-Selected

This property identifies the image to be displayed when the Check-Box control is selected. The number

corresponds to the position occupied by the image in the bitmap strip.

Example - Define a checkbox in screen section and define what bitmap to use when the check box is selected

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "My Check Box"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   .
```

Bitmap-Width

This property identifies the width in pixels of the image displayed in the Check-Box control. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

Example - Define a checkbox in screen section and define the width of each bitmap on a image containing many

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   flat  
   title "My Check Box"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Check-Box control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Check-Box control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Check-Box control will be relative to the ending position of

the prior Screen Section item.

When the Check-Box control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Check-Box, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a checkbox at column 5.0 on the screen section definition screen section

```
...  
03 screen-1-br-1 Check-Box  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Check-Box control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a check-box in screen section with a specific color

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   color 260  
   id 10  
   title "Sugar with the coffee?"  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a check-box in screen section with CSS base style name, applicable for EIS WD2 only

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   css-base-style-name "css-cb-style"  
   id 10  
   title "Sugar with the coffee?"  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a check-box in screen section with CSS style name, applicable for EIS WD2 only

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   css-style-name "css-cb-style"  
   id 10  
   title "Sugar with the coffee?"  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a checkbox

```
procedure division.  
...  
   modify screen-1-cb-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Check-Box control is disabled, "1" if it is enabled.

Example - Define a check-box in screen section, initially disabled and then enable it in procedure division

```
working-storage section.
...
01 filler pic 9 value 0.
   88 drinking-coffee value 1 false 0.

screen section.
...
03 screen-1-cb-1 Check-Box
   line 22.0
   column 10.0
   size 11.0 cells
   lines 3.0 cells
   enabled 0
   id 10
   title "Sugar with the coffee?"
   .
...
procedure division.
...
   if drinking-coffee
       modify screen-1-cb-1 enabled 1
   end-if.
...
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the `isgui.def` copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a check-box in screen section and exclude events to prevent them firing

```
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 116
   lines 31
   id 11
   event-list ( cmd-clicked msg-validate)
   exclude-event-list 1
   flat
   title "My Check Box"
   .
```

Exception-Value

If a numeric value different from "0" is set for this property and the [Notify](#) style is set, an exception condition for the active screen is generated when the value of the Check-Box control is modified.

Example - Define a check-box in screen section that raises an exception when it is clicked

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   exception-value 2090  
   line 27.7  
   column 10.6  
   size 116  
   lines 31  
   id 11  
   notify  
   .
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a check-box in screen section and define the only events to fire for it

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   event-list ( cmd-clicked msg-validate)  
   exclude-event-list 0  
   flat  
   title "My Check Box"  
   .
```

Font

This property specifies the font used to display the content of the Check-Box control. It may be used to compute the height and the width of the Check-Box control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a check-box in screen section with specific font

```
working-storage section.
...
77 Courier-New-0v0 handle of font.
...
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.7
   column 10.6
   size 156
   lines 31
   font Courier-New-0v0
   id 11
   .
...
procedure division.
...
   initialize wfont-data courier-new-0v0.
   move 0 to wfont-size.
   move "Courier New" to wfont-name.
   set wfont-bold to false.
   set wfont-italic to false.
   set wfont-underline to false.
   set wfont-strikeout to false.
   set wfont-fixed-pitch to false.
   call "w$font" using wfont-get-font courier-new-0v0
   wfont-data.
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Check-Box control. See "[Color management](#)" for further details.

Example - Define a check-box with a specific foreground color

```
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 156
   lines 31
   foreground-color 13
   id 11
   .
```

Help-Id

This property allows you to assign a unique ID to the Check-Box control to be passed to the help processor. See [Help automation](#) for more information.

Example - Define a check-box with a help id

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 156  
   lines 31  
   help-id 50  
   id 11  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Check-Box control.

Note - If the **Bitmap** style is set, no **Title-Position** has been specified and both **Title** and **Hint** have been set, then the **Title** text is used for the tool-tip.

Example - Define a check-box with a hint text

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 156  
   lines 31  
   hint "Press this check box to accept the above terms"  
   id 11  
   .
```

Id

This property allows you to assign a unique ID to the Check-Box control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in *isrct.def*.

Example - Define a check-box with a specific Id

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 156  
   lines 31  
   id 11  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a check-box that allows resize on X and Y when the window is resizing when a layout manager is defined for the screen

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 156  
   lines 31  
   id 11  
   layout-data 17 |*> To allow X and Y resizing  
   .
```

Left-Text-Alignment

This property works in conjunction with the [Left-Text](#) style and controls the alignment of the text in the area on the left of the Check-Box. Set this property to 0 to have the text right aligned or to 1 to have the text left aligned. If this property is not set, then the text is right aligned.

Example - Define a check-box with the text right aligned

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 156  
   lines 31  
   id 11  
   title "My Check Box"  
   left-text-alignment 0  
   .
```

Line

This property allows you to specify the Check-Box control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Check-Box control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Check-Box control will be relative to the starting position of the prior Screen Section item.

When the Check-Box control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Check-Box, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a check-box at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Check-Box control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Check-Box control is still computed in CELLS, but the cell size is based on the font set for the Check-Box control with the [Font](#) property. If no font has been defined for the Check-Box control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

If the BITMAP style is set, Lines are measured in pixels by default.

Example - Define a check-box with a height in lines

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 156  
   lines 31  
   id 11  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior

Example - Define a check-box with a maximum height if getting resized by layout manager

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   id 10  
   max-width 20.0  
   min-width 12.0  
   min-height 1.0  
   max-height 4.0  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a check-box with a maximum width if getting resized by layout manager

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   id 10  
   max-width 20.0  
   min-width 12.0  
   min-height 1.0  
   max-height 4.0  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a check-box with a minimum height if getting resized by layout manager

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   id 10  
   max-width 20.0  
   min-width 12.0  
   min-height 1.0  
   max-height 4.0  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a check-box with a minimum width if getting resized by layout manager

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   id 10  
   max-width 20.0  
   min-width 12.0  
   min-height 1.0  
   max-height 4.0  
   .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Check-Box control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a check-box with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   pop-up menu hmenu  
   line 4.6  
   column 9.0  
   size 10.0 cells  
   lines 3.7 cells  
   id 2  
   title "Check-Box"  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Check-Box control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Check-Box control is still computed in CELLS, but the cell size is based on the font set for the Check-Box control with the [Font](#) property. If no font has been defined for the Check-Box control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

If the **BITMAP** style is set, Size is measured in pixels by default.

Example - Define a check-box with a size

```
screen section.  
...  
03 screen-1-cb-1 Check-Box  
   line 22.0  
   column 10.0  
   size 11.0 cells  
   lines 3.0 cells  
   id 10  
   .
```

Termination-Value

If this property is set to a numeric value different from "0" and the [Notify](#) style is set, a termination condition for the active screen is generated when the value of the Check-Box control is modified.

Example - Define a check-box in screen section that raises a termination value when it is clicked

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   termination-value 2090  
   line 27.7  
   column 10.6  
   size 116  
   lines 31  
   id 11  
   notify  
   .
```

Title

The description shown in the Check-Box control. If the **Bitmap** style is set, no text is shown and the title becomes the control hint. If the **Title-Position** property is set, both text and graphics are shown.

Note - If the **Bitmap** style is set, no **Title-Position** has been specified and both **Title** and **Hint** have been set, then the **Title** text is used for the tool-tip.

Example - Define a check-box in screen section with a title

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   title "E-mail Results?"  
   title-position 2  
   .
```

Title-Position

This property assigns the position of the title of the Check-Box control when the **Bitmap** style is set; graphics and text are combined. Valid values are:

- | | |
|---|---------------------------|
| 1 | On the left of the image |
| 2 | On the right of the image |
| 3 | Above the image |
| 4 | Below the image |

Example - Define a check-box in screen section with a title on the right side

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   title "E-mail Results?"  
   bitmap-handle my-cb-png  
   bitmap-number 1  
   bitmap-width 20  
   title-position 2  
   bitmap-selected 2  
   title-position 2  
   .
```

Value

This property represents the value of the Check-Box control.

When inquired, it returns the value that is currently represented.

When set, the Check-Box control changes its look to represent it.

When set to zero, the Check-Box control is unchecked, when set to any other value, the Check-Box control is checked.

Example - Define a check-box in screen section with an initial value of not selected

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   title "E-mail Results?"  
   value 0  
   .
```

Visible

This property assumes a value of "0" if the Check-Box control is not visible, "1" if it is visible.

Example - Define a check-box in screen section that is not visible and will appear during procedure division

```
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 116
   lines 31
   id 11
   title "Sugar for you coffee?"
   visible 0
   .
...
procedure division.
...
   if drinking-coffee
       modify screen-1-cb-2 visible 1
   end-if.
...
```

Styles

The following styles are applicable to the CHECK-BOX control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bitmap](#), [Bold](#), [Flat](#), [Framed](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Left-Text](#), [Low](#), [Lowlight](#), [Multiline](#), [No-Tab](#), [Notify](#), [Permanent](#), [Self-Act](#), [Square](#), [Standard](#), [Temporary](#), [Transparent](#), [Unframed](#), [Vtop](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a check-box in screen section with high background

```
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 116
   lines 3.0
   id 11
   title "Sugar for you coffee?"
   background-high
   .
```

Bitmap

The Check-Box control is rendered like a [bitmap Push-Button](#). It appears pressed when the Check-Box control is checked.

The [Bitmap-Handle](#) property must be set.

When the [Title-Position](#) property is set, both the graphics and the title are drawn. Otherwise, the title becomes the hint of the control.

Example - Define a check-box in screen section with style bitmap

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   title "Sugar for you coffee?"  
   bitmap  
   .
```

Flat

When this style is set, the Check-Box control has no 3-D effect. When the mouse pointer is moved over the Check-Box control, it is highlighted.

Example - Define a check-box in screen section with style flat

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 31  
   id 11  
   title "Sugar for you coffee?"  
   flat  
   .
```

{ Framed | Unframed }

These styles have no effect, they're supported for compatibility with other COBOLs.

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Check-Box control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value](#) CELLS".

Example - Define a check-box in screen section with its height in cells

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 3.0  
   id 11  
   title "Sugar for you coffee?"  
   height-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a check-box in screen section with bold foreground

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   bold  
   .
```

Left-Text

When this style is set, the title is displayed on the left side. You can set the alignment of the title text through the property [Left-Text-Alignment](#).

Example - Define a check-box in screen section with text on the left side

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   left-text  
   .
```

Multiline

When this style is set, the title can be displayed on multiple lines. This happens when the title does not fit the size of the Check-Box control or when it contains a LineFeed character (x"0A").

If the `Lines` property is set to any value, the Multiline style is implied.

Example - Define a check-box in screen section with text on multiple lines

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Are you sure you would like some extra sugar for you coffee?"  
   multiline  
   .
```

No-Tab

Controls with this style set are skipped when the user navigates the screen using the Tab and the Backtab keys.

Example - Define a check-box in screen section that would be skipped when tab is pressed

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   no-tab  
   .
```

Notify

This style causes a **CMD-CLICKED** event to be a terminating event. The ACCEPT statement will terminate when the user changes the Check-Box value.

Example - Define a check-box in screen section with the notify style

```
screen section.
...
03 screen-1-cb-2 Check-Box
   line 27.0
   column 10.0
   size 116
   lines 30
   id 11
   event procedure screen-1-cb-2-evt-proc
   title "Sugar for you coffee?"
   notify
   .
...
procedure division.
...
screen-1-cb-1-evt-proc.
   evaluate event-control-id
   when 11
      evaluate event-type
      when cmd-clicked
         display message "The check-box [Sugar for your coffee] was clicked"
      when other
         end-evaluate
      end-evaluate
   end-evaluate.
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary check-box in screen section

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   temporary  
   .
```

Self-Act

When this style is set, all the events the Check-Box control fires are trapped and no Event Procedure is started. If either the [Exception-Value](#) property or the [Termination-Value](#) property is set, the ACCEPT Statement terminates with an Exception or Termination value.

Example - Define a check-box in screen section that produces no events but an exception

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   exception-value 5025  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   self-act  
   title "Sugar for you coffee?"  
   .
```

Square

Treated as a comment. The compiler recognizes this style for compatibility reasons.

Transparent

When this style is set, the title background becomes transparent.

Example - Define a check-box with transparent title background

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   transparent  
   .
```

Vtop

When this style is not set, the default, the title is vertically centered. When it is set, the title is aligned to the top.

Example - Define a check-box in screen section with title aligned at the top of the vertical space

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 116  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   vtop  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Check-Box control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a check-box with width in cells

```
screen section.  
...  
03 screen-1-cb-2 Check-Box  
   line 27.0  
   column 10.0  
   size 11.0  
   lines 30  
   id 11  
   title "Sugar for you coffee?"  
   width-in-cells  
   .
```

Events

The following events are applicable to the CHECK-BOX control: [CMD-CLICKED](#), [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#).

CMD-CLICKED

This event is fired when the Check-Box control is clicked. If the [Notify](#) style is set, this event terminates the ACCEPT.

CMD-GOTO

This event is fired when the user tries to activate the Check-Box control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Check-Box control is requested. The EVENT-DATA-2 data item contains the Check-Box control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

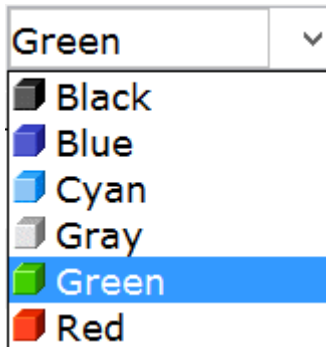
This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

COMBO-BOX



A Combo-Box combines an Entry-Field and a list that can be used to pick a value. Depending on the styles set, the Entry-Field can be editable or not.

The list is open by a left mouse click, by pressing F4 or by pressing ALT along with either the up arrow key or the down arrow key.

Properties

The following properties are applicable to the COMBO-BOX control: [Background-Color](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Cursor](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exception-Value](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hidden-Data](#), [Hint](#), [Id](#), [Insertion-Index](#), [Item](#), [Item-Text](#), [Item-To-Add](#), [Item-To-Delete](#), [Item-Value](#), [Layout-data](#), [Line](#), [Lines](#), [Mass-Update](#), [Max-Height](#), [Max-Text](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Placeholder](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Query-Index](#), [Reset-List](#), [Size](#), [Termination-Value](#), [Value](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Combo-Box control. See "[Color management](#)" for further details.

Example - Define a combo-box in screen section with dark cyan background color

```
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.0
   column 23.0
   size 12.0 cells
   lines 11.0
   background-color 3
   id 13
   3-d
   drop-down
   .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used.

Example - Define a combo-box that uses bitmap images for the items

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.4
   column 23.2
   size 11.9 cells
   lines 11.1
   id 13
   3-d
   drop-down
   bitmap-width 20
   bitmap-handle icon-png
   .
...
procedure division.
...
   *> Load the bitmap to a handle
   call "w$bitmap"
       using wbitmap-load "icon.png" giving icon-png.
...
   *> Add 3 items and assign a bitmap image to each
   modify screen-1-co-1 item-to-add ("Blue", "Cyan", "Green")
   modify screen-1-co-1(1) bitmap-number 20
   modify screen-1-co-1(2) bitmap-number 16
   modify screen-1-co-1(3) bitmap-number 17.
...
```

Bitmap-Number

This property identifies the image to be displayed on the side of the item identified by the [Item](#) property.

Example - Define a combo-box that uses bitmap images for the items

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.4
   column 23.2
   size 11.9 cells
   lines 11.1
   id 13
   3-d
   drop-down
   bitmap-width 20
   bitmap-handle icon-png
   .
...
procedure division.
...
   *> Load the bitmap to a handle
   call "w$bitmap"
       using wbitmap-load "icon.png" giving icon-png.
...
   *> Add 3 items and assign a bitmap image to each
   modify screen-1-co-1 item-to-add ("Blue", "Cyan", "Green")
   modify screen-1-co-1(1) bitmap-number 20
   modify screen-1-co-1(2) bitmap-number 16
   modify screen-1-co-1(3) bitmap-number 17.
...
```

Bitmap-Width

This property identifies the width in pixels of the image displayed on the side of the item identified by the [Item](#) property. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

Example - Define a combo-box that uses bitmap images for the items

```
working-storage section.
...
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.4
   column 23.2
   size 11.9 cells
   lines 11.1
   id 13
   3-d
   drop-down
   bitmap-width 20
   bitmap-handle icon-png
   .
...
procedure division.
...
   *> Load the bitmap to a handle
   call "w$bitmap"
       using wbitmap-load "icon.png" giving icon-png.
...
   *> Add 3 items and assign a bitmap image to each
   modify screen-1-co-1 item-to-add ("Blue", "Cyan", "Green")
   modify screen-1-co-1(1) bitmap-number 20
   modify screen-1-co-1(2) bitmap-number 16
   modify screen-1-co-1(3) bitmap-number 17.
...
```

[Col | Column | Pos | Position]

This property allows you to specify the Combo-Box control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Combo-Box control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Combo-Box control will be relative to the ending position of the prior Screen Section item.

When the Combo-Box control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).
03 Combo-Box, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a combo-box at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Combo-Box control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a combo-box in screen section with grey background and light grey foreground all in the color property

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   color 265  
   id 13  
   3-d  
   drop-down  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a combo-box in screen section with a css-base-style-name, valid with EIS WD2

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   color 265  
   id 13  
   css-base-style-name "cb-style"  
   3-d  
   drop-down  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a combo-box in screen section with a css-style-name, valid with EIS WD2

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   color 265  
   id 13  
   css-style-name "cb-style"  
   3-d  
   drop-down  
   .
```

Cursor

This property set or retrieves the cursor position inside an entry-field part of a [Drop-Down](#) combo-box. If the value "-1" is assigned to this property, the whole text in the entry-field is selected and the cursor is positioned at the end of it.

Example - Define a combo-box in screen section and define the cursor to position and the end of the entry-

field part and select all value

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   color 265  
   id 13  
   cursor -1  
   3-d  
   drop-down  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a combo-box

```
procedure division.  
...  
   modify screen-1-co-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Combo-Box control is disabled, "1" if it is enabled.

Example - Define a disabled combo-box in screen section and enable it later in procedure division

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   color 265  
   enabled 0  
   id 13  
   cursor -1  
   3-d  
   drop-down  
   .  
...  
procedure division.  
...  
   if enable-combo-box  
       modify screen-1-co-1 enabled 1  
   end-if  
   .  
...  
...
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a combo-box in screen section and define a list of events to be excluded

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   enabled 0  
   id 13  
   event-list ( cmd-dblclick msg-validate)  
   exclude-event-list 1  
   .  
...  
...
```

Exception-Value

If a numeric value different from "0" is set for this property and the [Notify-Selchange](#) style is set, an exception condition for the active screen is generated when the value of the Combo-Box control is modified.

Example - Define a combo-box with an exception value

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   exception-value 100  
   id 13  
   .  
...
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a combo-box in screen section and define a list of events to be excluded

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.4  
   column 23.2  
   size 11.9 cells  
   lines 11.1  
   enabled 0  
   id 13  
   event-list ( cmd-dblclick msg-validate)  
   exclude-event-list 1  
   .  
...
```

Font

This property specifies the font used to display the content of the Combo-Box control. It may be used to compute the height and the width of the Combo-Box control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), [Width-In-Cells](#) properties for further details.

Example - Define a combo-box in screen section with font Tahoma

```
working-storage section.
...
copy "isfonts.def".
77 Tahoma-8v0 handle of font.
...
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.4
   column 23.2
   size 11.9 cells
   lines 11.1
   font Tahoma-8v0
   id 13
   .
...
procedure division.
...
   initialize wfont-data tahoma-8v0.
   move 8 to wfont-size.
   move "Tahoma" to wfont-name.
   set wfont-bold to false.
   set wfont-italic to false.
   set wfont-underline to false.
   set wfont-strikeout to false.
   set wfont-fixed-pitch to false.
   call "w$font" using wfont-get-font tahoma-8v0 wfont-data.
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Combo-Box control. See "[Color management](#)" for further details.

Example - Define a combo-box in screen section with dark cyan foreground color

```
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.0
   column 23.0
   size 12.0 cells
   lines 11.0
   foreground-color 3
   id 13
   3-d
   drop-down
   .
```

Help-Id

This property allows you to assign a unique ID to the Combo-Box control to be passed to the help processor. See [Help automation](#) for more information.

Example - Define a combo-box in screen section with help-id

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   foreground-color 3  
   help-id 510  
   id 13  
   3-d  
   drop-down  
   .
```

Hidden-Data

This property can be used to change the hidden data of an item. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to an item.

The `Item` property must be set first. Setting this property before the `Item` property is not advised as it may lead to unexpected behaviors.

Example - Add some items to a combo-box including hidden data for each

```
...  
procedure division.  
...  
   modify screen-1-co-1  
       item-to-add "Grey"  
       hidden-data "i"  
       item-to-add "Black"  
       hidden-data "ii"  
       item-to-add "White"  
       hidden-data "iii".
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Combo-Box control.

Example - Create a combo-box in screen section with hint text

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   hint "Select color here"  
   3-d  
   drop-down  
   .
```

Id

This property allows you to assign a unique ID to the Combo-Box control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in *isqrt.def*.

Example - Create a combo-box in screen section with an ID

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   .
```

Insertion-Index

This numeric property is used to set the position of the newly inserted item. The *Unsorted* style must be set; if not the items are alphabetically sorted. The new item is inserted right before the item number specified here. For example, with the value "1" the next item added to the Combo-Box control will appear at the top of the list, while with the value "0" the item will be inserted at the end of it.

Example - Add one item to a combo-box in 2nd position the combo-box should have the unsorted style

```
procedure division.  
...  
   modify screen-1-co-1 insertion-index 2 item-to-add "Yellow".  
...  
.
```

Item

This property identifies, through a number, a Combo-Box control item. The first item in the list is referenced

with "1", the second with "2" and so on. This property must be set before the [Item-Text](#), [Hidden-Data](#), [Bitmap-Handle](#) and [Bitmap-Number](#) properties to change or inquire the text, hidden data or the image of that item.

Example - Modify the text of the third element of the combo-box to "Hello world" and the image with the fourth bitmap in the bitmap strip referenced by the handle MY_BITMAP.

```
modify screen-1-co-1
  item          = 3
  item-text     = "Hello World"
  bitmap-handle = my_bitmap
  bitmap-number = 4.
```

When inquired, this property returns a number specifying the currently selected list item or 0 if no item is selected.

Item-Text

This property can be used to change the text of an item.

The [Item](#) property must be set first. Setting this property before the [Item](#) property is not advised as it may lead to unexpected behaviors.

Example - Modify a combo-box text for an item

```
procedure division.
...
  modify screen-1-co-1 item 3 item-text "Changed item".
...
```

Item-To-Add

When a value is assigned to this property, a new item is added to the list.

Multiple values can be added at the same time, enclosed between parentheses.

The position of the new item can be controlled by the [Insertion-Index](#) property, provided that the Combo-Box control has the [Unsorted](#) style set.

Example - Add a new item to the combo-box

```
procedure division.
...
  modify screen-1-co-1 item-to-add ("Blue", "Cyan", "Green").
...
```

Item-To-Delete

As soon as the value of this property is modified, the corresponding item in the list is removed.

Each item is identified by a number that matches its position in the list, starting at 1.

Example - Modify a combo-box to delete the 2nd item on it

```
...  
procedure division.  
...  
    modify screen-1-co-1 item-to-delete 2.  
...
```

Item-Value

Item-Value is synonymous with the [Item-Text](#) property.

Example - Modify a combo-box text for an item

```
procedure division.  
...  
    modify screen-1-co-1 item 3 item-value "Changed item".  
...
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a combo-box in screen section with layout data to resize X and Y from the layout manager when the screen gets resized

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
    line 19.0  
    column 23.0  
    size 12.0 cells  
    lines 11.0  
    id 13  
    layout-data 17  
    3-d  
    drop-down  
    .
```

Line

This property allows you to specify the Combo-Box control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Combo-Box control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Combo-Box control will be relative to the starting position of the prior Screen Section item.

When the Combo-Box control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).
03 Combo-Box, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a combo-box at line 8.0 on the screen section definition

```
screen section.
...
03 screen-1-br-1 Bar
   line 8.0
   column 5.0
   color 7
   size 45.0 cells
   id 2
   .
```

Lines

This property allows you to specify the height of the Combo-Box control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Combo-Box control is still computed in CELLS, but the cell size is based on the font set for the Combo-Box control with the [Font](#) property. If no font has been defined for the Combo-Box control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

The Lines property measures the height of the list that appears when the combo-box is expanded.

Example - Defined a combo-box in screen section specifying its height in lines

```
screen section.
...
03 screen-1-co-1 Combo-Box
   line 19.0
   column 23.0
   size 12.0 cells
   lines 11.0
   id 13
   3-d
   drop-down
   .
```

Mass-Update

Setting this property to "1" inhibits isCOBOL framework to repaint the Combo-Box control every time the program modifies it. This practice is recommended in order to increase performance when a large number of changes are applied to the Combo-Box control. At the end of the process it is necessary to reset the property to its default value of "0" to see the changes.

Example - Load three items into a combo-box under mass update

```
modify screen-1-co-1 mass-update = 1
modify screen-1-co-1 item-to-add "item 1"
modify screen-1-co-1 item-to-add "item 2"
modify screen-1-co-1 item-to-add "item 3"
modify screen-1-co-1 mass-update = 0
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a combo-box in screen section with a maximum height when the control gets resized by the layout manager

```
screen section.
...
03 screen-1-co-1 Combo-Box
  line 19.0
  column 23.0
  size 12.0 cells
  lines 11.0
  id 13
  max-height 20.0
  layout-data 17
  3-d
  drop-down
  .
```

Max-Text

This property can be used to set the maximum number of characters the user can enter. If this property is not specified, it is set to the same value as the [Size](#) property.

Example - Define a combo-box in screen section with a limit on the number of characters to enter

```
screen section.
...
03 screen-1-co-1 Combo-Box
  line 19.0
  column 23.0
  size 12.0 cells
  lines 11.0
  id 13
  max-text 15
  layout-data 17
  3-d
  drop-down
  .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a combo-box in screen section with a maximum width when the control gets resized by the layout manager

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   max-width 20.0  
   layout-data 17  
   3-d  
   drop-down  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a combo-box control that can be resized by the layout-manager and has a minimum height

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   min-height 20.0  
   layout-data 17  
   3-d  
   drop-down  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a combo-box control that can be resized by the layout-manager and has a minimum width

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   min-width 20.0  
   layout-data 17  
   3-d  
   drop-down  
   .
```

Placeholder

This property specifies a short hint that describes the expected value of an input field. The short hint is displayed in the input field before the user enters a value. It doesn't affect [Drop-List](#) combo-boxes.

The color of the placeholder text can be configured by setting `iscobol.gui.placeholder_color` in the configuration.

Note - The placeholder text is not necessarily displayed in the same position as the input text.

Example - Define a combo-box with a placeholder text

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   placeholder "Select the language"  
   .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Combo-Box control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a combo-box with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-co-1 Combo-Box  
  pop-up menu hmenu  
  line 19.0  
  column 23.0  
  size 12.0 cells  
  lines 11.0  
  id 13  
  max-width 20.0  
  layout-data 17  
  3-d  
  drop-down  
  .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Query-Index

This property is used in conjunction with the [Item-Value](#) and [Hidden-Data](#) properties to retrieve the value of a specific item and to manage the hidden data bind to the item.

Each item is identified by a number that corresponds to its position in the list, starting at 1.

Example - Get the value of a combo-box item

```
procedure division.  
...  
  modify screen-1-co-1 query-index 3  
  inquire screen-1-co-1 item-value my-item-var  
...  

```

Reset-List

By assigning a value other than zero to this property, all the items are removed from the Combo-Box control.

Example - Empty a combo-box

```
procedure division.  
...  
  modify screen-1-co-1 reset-list 1  
...  

```

Size

This property allows you to specify the size of the Combo-Box control. If the PIXEL keyword follows the value

specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Combo-Box control is still computed in CELLS, but the cell size is based on the font set for the Combo-Box control with the [Font](#) property. If no font has been defined for the Combo-Box control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a combo-box in screen section saying what its size is

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   .
```

Termination-Value

If this property is set to a numeric value different from "0" and the [Notify-Selchange](#) style is set, a termination condition for the active screen is generated when the value of the Combo-Box control is modified.

Example - Define a combo-box in screen section that produces a termination value when an item is selected

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   notify-selchange  
   3-d  
   drop-down  
   termination-value 5030  
   .
```

Value

This property represents the value of the Combo-Box control.

When inquired, it returns the value that is currently represented.

When set, the Combo-Box control changes its look to represent it.

When set in a [Drop-List](#) Combo-Box, the currently selected item is changed in order to match the value set. If set to an invalid value (a value that is not listed inside the control), then the Combo-Box control appears

empty.

Example - Define a combo-box with its default value

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   notify-selchange  
   3-d  
   drop-down  
   value "Item1"  
   termination-value 5030  
   .  
...
```

Visible

This property assumes a value of "0" if the Combo-Box control is not visible, "1" if it is visible.

Example - Define an invisible combo-box in screen section to make it visible later in procedure division

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   visible 0  
   .  
...  
procedure division.  
...  
  modify screen-1-co-1 item-to-add ( "Red", "Blue", "Green", "Black" ).  
...  
  if color-selection  
    modify screen-1-co-1 visible 1  
  end-if.  
...  
...
```

Styles

The following styles are applicable to the COMBO-BOX control: Background-High, Background-Low, Background-Standard, Bold, Drop-Down, Drop-List, Height-In-Cells, High, Highlight, Low, Lower, Lowlight, No-Autosel, No-Tab, Notify-Dbclick, Notify-Selchange, Permanent, Standard, Static-List, Temporary, Unsorted, Upper, Width-In-Cells.

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a combo-box with high background color

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   background-high  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   .
```

{ Drop-Down | Drop-List | Static-List }

Drop-Down	This style should be used when space for the Combo-Box is limited, for example when the combo box is part of a toolbar. With this style, the list, part of the Combo-Box, is normally hidden and shown only when the user clicks the button on the right of the entry area. The user can pick a value from the list or enter any value in the entry area.
Drop-List	This style has the same effect as the DROP-DOWN style, with the exception that the entry area is read-only.
Static-List	This syntax is compiled for compatibility reasons, however, isCOBOL does not support Static-List Combo Boxes.

Example - Define a combo-box with drop-down style

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Combo-Box control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a combo-box in screen section with height in cells

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   height-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a combo-box with bold foreground color

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
  bold  
  line 19.0  
  column 23.0  
  size 12.0 cells  
  lines 11.0  
  id 13  
  3-d  
  drop-down  
  .
```

{ Lower | Upper }

Lower	When this style is set, all the items are converted to lower-case characters.
Upper	When this style is set, all the items are converted to upper-case characters.

Example - Define a combo-box in screen section with uppercase style

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
  line 19.0  
  column 23.0  
  size 12.0 cells  
  lines 11.0  
  id 13  
  upper  
  3-d  
  drop-down  
  .
```

No-Autosel

When this style is set, the content of the combo-box is not automatically selected when activated.

Example - Define a combo-box in screen section with no-autosel style

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   no-autosel  
   3-d  
   drop-down  
   .
```

No-Tab

Controls with this style set are skipped when the user navigates the screen using the Tab and the Backtab keys.

Example - Define a combo-box in screen section that is not navigable with tab key

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   no-tab  
   3-d  
   drop-down  
   .
```

Notify-Dbclick

This style causes a [CMD-DBLCLICK](#) event to be fired any time the user double-clicks on an item. Without this style, no event is generated under this circumstance.

Example - Define a combo-box in screen section that fires an event when it is double-clicked

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   notify-dblclick  
   3-d  
   drop-down  
   .
```

Notify-Selchange

This style causes a **NTF-SELCHANGE** event to be fired any time the user selects an item. Without this style, no event is generated under this circumstance.

Example - Define a combo-box in screen section that fires an event when the selection changes

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   notify-selchange  
   3-d  
   drop-down  
   .
```

{ Permanent | Temporary }

Permanent

A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.

Temporary

Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a combo-box in screen section with a temporary style

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   temporary  
   3-d  
   drop-down  
   .
```

Unsorted

When this style is set, items contained in the list are shown in the same order they have been added to the Combo-Box control.

Example - Define a combo-box in screen section that does not sort items automatically

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   unsorted  
   3-d  
   drop-down  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Combo-Box control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a combo-box in screen section with its width in cells

```
screen section.  
...  
03 screen-1-co-1 Combo-Box  
   line 19.0  
   column 23.0  
   size 12.0  
   lines 11.0  
   id 13  
   width-in-cells  
   3-d  
   drop-down  
   .
```

Events

The following events are applicable to the COMBO-BOX control: [CMD-DBLCLICK](#), [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#), [NTF-SELCHANGE](#).

CMD-DBLCLICK

This event is fired when the user double-clicks on an item of a combo or list box and either the [Termination-Value](#) property or the [Exception-Value](#) property is set. The EVENT-DATA-1 data item contains the index associated with the selected item.

CMD-GOTO

This event is fired when the user tries to activate the Combo-Box control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Combo-Box control is requested. The EVENT-DATA-2 data item contains the Combo-Box control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value.

This is used when the programmer wants to handle menu actions in the Event Procedure.

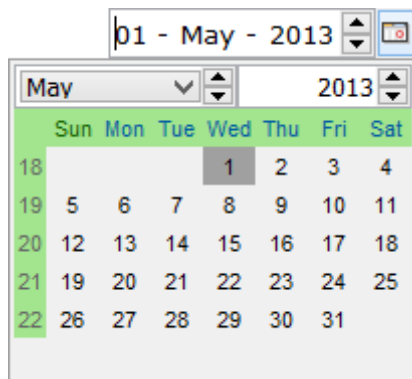
MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting `EVENT-ACTION` to `EVENT-ACTION-CONTINUE` causes the focus to remain on the control, allowing the user to correct errors.

NTF-SELCHANGE

This event is fired when the user selects a new item in a combo or list box created with the [Notify-Selchange](#) style. The `EVENT-DATA-1` data item contains the selected item ID.

DATE-ENTRY



A Date-Entry is a special Entry-Field that allows you to display and enter date or time information. A button on its right side allows the user to pop-up a calendar to pick dates.

The calendar's appearance is conditioned by the international settings of the machine that uses it: months and days names are translated to the local language and the displayed date format respects the international settings (unless you set [Display-Format](#) to a precise date format).

Properties

The following properties are applicable to the DATE-ENTRY control: [Background-Color](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), [Calendar-Font](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Decoration-Background](#), [Display-Format](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Val](#), [Max-Width](#), [Maxday-Characters](#), [Min-Height](#), [Min-Val](#), [Min-Width](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Size](#), [Sunday-Foreground](#), [Value](#), [Value-Format](#), [Visible](#), [Weekday-Foreground](#).

Background-Color

This property allows you to set or retrieve the background color of the Date-Entry control. See "[Color management](#)" for further details.

Example - Define a date-entry control with yellow background and brown foreground colors

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   background-color 14  
   foreground-color 6  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used for the button that opens the calendar.

Example - Define a date-entry control with specific bitmap image for the calendar button

```
working-storage section.  
77 icon-png1 pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   bitmap-number 11  
   bitmap-width 18  
   bitmap-handle icon-png1.  
...  
procedure division.  
...  
call "w$bitmap" using wbitmap-load "icon.png" giving  
   icon-png1.  
...
```

Bitmap-Number

This property defines which bitmap among the ones in the bitmap strip referenced by the [Bitmap-Handle](#) property is to be used for the button that opens the calendar.

Example - Define a date-entry control with specific bitmap image for the calendar button

```
working-storage section.  
77 icon-png1 pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   bitmap-number 11  
   bitmap-width 18  
   bitmap-handle icon-png1.  
...  
procedure division.  
...  
call "w$bitmap" using wbitmap-load "icon.png" giving  
   icon-png1.  
...
```

Bitmap-Width

This property defines the width in pixels of the image used for the button that opens the calendar.

Example - Define a date-entry control with specific bitmap image for the calendar button

```
working-storage section.  
77 icon-png1 pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   bitmap-number 11  
   bitmap-width 18  
   bitmap-handle icon-png1.  
...  
procedure division.  
...  
call "w$bitmap" using wbitmap-load "icon.png" giving  
   icon-png1.  
...
```

Calendar-Font

This property specifies the font used by the pop-up calendar.

Example - Define a date-entry control with specific calendar-font

```
working-storage section.  
copy "isfonts.def".  
77 Aharoni-10v0 handle of font.  
...  
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyyymmdd  
   calendar-font Bell-MT-10v0-b  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Date-Entry control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Date-Entry control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Date-Entry control will be relative to the ending position of the prior Screen Section item.

When the Date-Entry control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Date-Entry, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a date-entry at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Date-Entry control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a date-entry control with single color property

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   color 230  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a date-entry control with css base style name, valid for EIS WD2

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   css-base-style-name "date-entry-style"  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a date-entry control with css style name, valid for EIS WD2

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   css-style-name "date-entry-style"  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a date-entry

```
procedure division.  
...  
   modify screen-1-de-1 custom-data "Screen 1 custom data"  
...  

```

Decoration-Background

This property allows you to set or retrieve the color of the column and row headings of the calendar. See "[Color management](#)" for further details.

Example - Define a date-entry control with decoration background color

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   decoration-background 12  
   .
```

Display-Format

This property allows you to set the pattern string that will be used to display the information. Within date and time pattern strings, unquoted letters from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters representing the components of a date or time string. Text can be quoted using single quotes (') to avoid

interpretation. "" represents a single quote. All other characters are not interpreted, they are simply copied into the output string during formatting or matched against the input string during parsing.

The following pattern letters are defined (all other characters from 'A' to 'Z' and from 'a' to 'z' are reserved):

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Pattern letters are usually repeated, as their number determines the exact presentation:

Text	For formatting, if the number of pattern letters is 4 or more, the full form is used; otherwise a short or abbreviated form is used if available. For parsing, both forms are accepted, independently from the number of pattern letters.
Number	For formatting, the number of pattern letters is the minimum number of digits and shorter numbers are zero-padded to this amount. For parsing, the number of pattern letters is ignored unless it is needed to separate two adjacent fields.

Year	<p>For formatting, if the number of pattern letters is 2, the year is truncated to 2 digits; otherwise it is interpreted as a number.</p> <p>For parsing, if the number of pattern letters is more than 2, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.</p> <p>For parsing with the abbreviated year pattern ("y" or "yy"), SimpleDateFormat must interpret the abbreviated year relative to some century. It does this by adjusting dates to be within 80 years before and 20 years after the time the SimpleDateFormat instance is created. For example, using a pattern of "MM/dd/yy" and a SimpleDateFormat instance created on Jan 1, 1997, the string "01/11/12" would be interpreted as Jan 11, 2012 while the string "05/04/64" would be interpreted as May 4, 1964. During parsing, only strings consisting of exactly two digits, as defined by Character.isDigit(char), will be parsed into the default century. Any other numeric string, such as a one digit string, a three or more digit string, or a two digit string that is not all digits (for example, "-1"), is interpreted literally. So, "01/02/3" or "01/02/003" are parsed as "Jan 2, 3 AD", using the same pattern. Likewise, "01/02/-3" is parsed as "Jan 2, 4 BC".</p>
Month	<p>If the number of pattern letters is 3 or more, the month is interpreted as text; otherwise, it is interpreted as a number.</p>
General time zone	<p>Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used:</p> <p>GMTOffsetTimeZone: GMT Sign Hours : Minutes</p> <p>Sign: one of + -</p> <p>Hours: Digit Digit Digit</p> <p>Minutes: Digit Digit</p> <p>Digit: one of 0 1 2 3 4 5 6 7 8 9</p> <p>Hours must be between 0 and 23, and Minutes must be between 00 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.</p> <p>For parsing, RFC 822 time zones are also accepted.</p>

RFC 822 time zone	<p>For formatting, the RFC 822 4-digit time zone format is used:</p> <p>RFC822TimeZone: Sign TwoDigitHours Minutes</p> <p>TwoDigitHours: Digit Digit</p> <p>TwoDigitHours must be between 00 and 23. Other definitions are as for general time zones.</p> <p>For parsing, general time zones are also accepted.</p>
-------------------	---

Localized date and time pattern strings are also supported. In these strings, the pattern letters described above may be replaced with other, locale dependent, pattern letters.

If this property is not set, then the display format is retrieved by the current operating system international settings.

Example - Define a date-entry control with display format that include era designator, year, month and day

```
screen section.
...
03 screen-1-de-1 Date-Entry
   line 21.2
   column 49.5
   size 14.5 cells
   lines 3.1 cells
   id 17
   century-date
   decoration-background-visible
   display-format "G, y-MM-d"
   value-format davf-yyyymmdd
   calendar-font Bell-MT-10v0-b
   decoration-background 12
   .
```

Enabled

This property assumes a value of "0" if the Date-Entry control is disabled, "1" if it is enabled.

Example - Define a date-entry control initially enabled to be disabled later on procedure division

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   enabled 1  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   .  
...  
procedure division.  
...  
   if disable-dates  
       modify screen-1-de-1 enabled 0  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a date-entry control with an event list to be excluded

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   enabled 0  
   id 17  
   event-list ( msg-validate ntf-changed)  
   exclude-event-list 1  
   century-date  
   value-format davf-yyyymmdd  
   .  

```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a date-entry control with an event list to be excluded

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
    line 21.2  
    column 49.5  
    size 14.5 cells  
    lines 3.1 cells  
    enabled 0  
    id 17  
    event-list ( msg-validate ntf-changed)  
    exclude-event-list 1  
    century-date  
    value-format davf-yyyymmdd  
    .
```

Font

This property specifies the font used to display the content of the Date-Entry control. It may be used to compute the height and the width of the Date-Entry control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a date-entry control with specific font

```
working-storage section.
copy "isfonts.def".
77 Aharoni-10v0 handle of font.
...
screen section.
...
03 screen-1-de-1 Date-Entry
   line 21.2
   column 49.5
   size 12.6 cells
   lines 3.1 cells
   font Aharoni-10v0
   id 17
   century-date
   value-format davf-yyyymmdd
   calendar-font Default-Font
   .
...
procedure division.
...
   initialize wfont-data aharoni-10v0.
   move 10 to wfont-size.
   move "Aharoni" to wfont-name.
   set wfont-bold to false.
   set wfont-italic to false.
   set wfont-underline to false.
   set wfont-strikeout to false.
   set wfont-fixed-pitch to false.
   call "w$font" using wfont-get-font aharoni-10v0 wfont-data.
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Date-Entry control. See "[Color management](#)" for further details.

Example - Define a date-entry control with yellow background and brown foreground colors

```
screen section.
...
03 screen-1-de-1 Date-Entry
   line 21.2
   column 49.5
   size 12.6 cells
   lines 3.1 cells
   id 17
   century-date
   value-format davf-yyyymmdd
   .
```

Help-Id

This property allows you to assign a unique ID to the Date-Entry control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a date-entry control with a Help-id

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 12.6 cells  
   lines 3.1 cells  
   help-id 4040  
   id 17  
   century-date  
   value-format davf-yyyyymmdd  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Date-Entry control.

Example - Define a date-entry control with a Hint text

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   hint "Enter your enrollment date"  
   century-date  
   value-format davf-yyyyymmdd  
   .
```

Id

This property allows you to assign a unique ID to the Date-Entry control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a date-entry control with an ID

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a date-entry control that gets resized in X and Y when the layout-manager requires so

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   layout-data 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Line

This property allows you to specify the Date-Entry control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Date-Entry control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Date-Entry control will be relative to the starting position of the prior Screen Section item.

When the Date-Entry control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Date-Entry, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a date-entry at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 8.0  
   column 5.0  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Date-Entry control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Date-Entry control is still computed in CELLS, but the cell size is based on the font set for the Date-Entry control with the [Font](#) property. If no font has been defined for the Date-Entry control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Defined a date-entry in screen section specifying its height in lines

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 19.0  
   column 23.0  
   size 12.0 cells  
   lines 11.0  
   id 13  
   3-d  
   drop-down  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a date-entry control that can be resized by the layout-manager and has a maximum height

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   help-id 4040  
   id 17  
   max-height 50.0  
   layout-data 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Max-Val

With this property it is possible to set the maximum allowed value. All days exceeding this value will not be enabled.

Example - Define a date-entry control with a maximum value

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   max-val "20150515"  
   value-format davf-yyyymmdd.
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a date-entry control that can be resized by the layout-manager and has a maximum width

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   help-id 4040  
   id 17  
   max-width 50.0  
   layout-data 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a date-entry control that can be resized by the layout-manager and has a minimum height

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   help-id 4040  
   id 17  
   min-height 10.0  
   layout-data 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a date-entry control that can be resized by the layout-manager and has a minimum width

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   help-id 4040  
   id 17  
   min-width 10.0  
   layout-data 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Maxday-Characters

This property affects the width of the columns and the number of characters for show the day name in the header of the calendar. It ranges from 1 to 3.

Example - Define an entry-field that shows only 1 character in the heading of each day column

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   maxday-characters 1.
```

Min-Val

With this property it is possible to set the minimum allowed value. All days below this value will not be enabled.

Example - Define a date-entry control with a minimum value

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   max-val "20150515"  
   min-val "20140515"  
   value-format davf-yyyymmdd.
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Date-Entry control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a date-entry with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   pop-up menu hmenu  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   max-val "20150515"  
   min-val "20140515"  
   value-format davf-yyyymmdd.  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Date-Entry control. If the [PIXEL](#) keyword follows the value specified here, the size is computed in pixels. If either the [CELLS](#) keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Date-Entry control is still computed in CELLS, but the cell size is based on the font set for the Date-Entry control with the [Font](#) property. If no font has been defined for the Date-Entry control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a date-entry with specific size

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
    line 21.2  
    column 49.5  
    size 14.5 cells  
    lines 3.1 cells  
    id 17  
    century-date  
    value-format davf-yyyymmdd.
```

Sunday-Foreground

This property allows you to set or retrieve the Sunday color. The default color is red. See "[Color management](#)" for further details.

Example - Define a date-entry control with yellow heading for sunday

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
    line 21.2  
    column 49.5  
    size 14.5 cells  
    lines 3.1 cells  
    id 17  
    century-date  
    value-format davf-yyyymmdd  
    sunday-foreground 15  
    .
```

Value

This property represents the value of the Date-Entry control.

When inquired, it returns the value that is currently represented.

When set, the Date-Entry control changes its look to represent it.

Example - Query the value of a date-entry control

```
...  
procedure division.  
...  
    inquire screen-1-de-1 value ws-birth-date  
...  

```

Value-Format

The information that the [Value](#) property refers to is a numeric data item. It does not contain information about its format. In other words, examining the [Value](#) property, there is no way to guess what digits represent the year, what the month and so on. Moreover, when the user enters a date, its elements (year, month, day) must be combined, in order to obtain a number to be assigned to the [Value](#) property.

The VALUE-FORMAT property fills that gap and allows you to convert information properly. Supported values, defined in [isgui.def](#) are:

DAVF-YYYYMMDD	Only the eight most significant digits are used. The first four digits are used for the year, the following two for the month and the last two for the day.
DAVF-YYMMDD	Only the six most significant digits are used. The first two digits are used for the year, the following two for the month and the last two for the day.
DAVF-MMDDYYYY	Only the eight most significant digits are used. The first two digits are used for the month, the following two for the day and the last four for the year.
DAVF-MMDDYY	Only the six most significant digits are used. The first two digits are used for the month, the following two for the day and the last two for the year.
DAVF-DDMMYYYY	Only the eight most significant digits are used. The first two digits are used for the day, the following two for the month and the last four for the year.
DAVF-DDMMYY	Only the six most significant digits are used. The first two digits are used for the day, the following two for the month and the last two for the year.
DAVF-HHMM	Only the four most significant digits are used. The first two digits are used for the hours and the following two for the minutes.
DAVF-HHMMSS	Only the six most significant digits are used. The first two digits are used for the hours, the following two for the minutes and the following two for the seconds. The Time style must be set.
DAVF-HHMMSShh	Only the eight most significant digits are used. The first two digits are used for the hours, the following two for the minutes, the following two for the seconds, and the last two for the hundredths of second. The Time style must be set.
DAVF-YYYYMMDDHHMMSShh	Only the sixteen most significant digits are used. The first eight digits are treated as described for DAVF-YYYYMMDD and the last eight are treated as described for DAVF-HHMMSShh. The Time style must be set.
DAVF-MMDDYYYYHHMMSShh	Only the sixteen most significant digits are used. The first eight digits are treated as described for DAVF-MMDDYYYY and the last eight are treated as described for DAVF-HHMMSShh. The Time style must be set.
DAVF-DDMMYYYYHHMMSShh	Only the sixteen most significant digits are used. The first eight digits are treated as described for DAVF-DDMMYYYY and the last eight are treated as described for DAVF-HHMMSShh. The Time style must be set.

Example - Define a date-entry control with value-format

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   .
```

Visible

This property assumes a value of "0" if the Date-Entry control is not visible, "1" if it is visible.

Example - Make a date-entry control become invisible during runtime

```
...  
procedure division.  
...  
   if dates-invisible  
       modify screen-1-de-1 visible 0  
   ...  
   end-if  
...
```

Weekday-Foreground

This property allows you to set or retrieve the color of the weekdays. See "[Color management](#)" for further details.

Example - Define a date-entry control with the weekday headings in magenta color

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Bell-MT-10v0-b  
   sunday-foreground 15  
   weekday-foreground 14  
   .
```


Styles

The following styles are applicable to the DATE-ENTRY control: [Allow-Empty](#), [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Century-Date](#), [Decoration-Background-Visible](#), [Decoration-Borders-Visible](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Long-Date](#), [Low](#), [Lowlight](#), [No-F4](#), [No-Tab](#), [No-Updown](#), [Notify-Change](#), [Numeric](#), [Permanent](#), [Read-Only](#), [Right-Align](#), [Self-Act](#), [Short-Date](#), [Show-None](#), [Spinner](#), [Standard](#), [Temporary](#), [Time](#), [Week-Of-Year-Visible](#), [Width-In-Cells](#).

Allow-Empty

When this style is set, the Date-Entry shows an empty field when the [Value](#) property is not set or is set to zero. Without this style, a date is always shown in the control.

Example - Define a date-entry control that allows an empty value

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   allow-empty  
   value-format davf-yyyymmdd  
   .
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a date-entry control with low background

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   background-low  
   century-date  
   value-format davf-yyyymmdd  
   .
```

{ Century-Date | Long-Date | Short-Date }

Century-Date	Same as Short-Date. The year has always four digits.
Long-Date	The date is displayed with the long date format.
Short-Date	The date is displayed with the short date format.

Date and time formats depend on the system configuration.

Example - Define a date-entry control with century-date to allow 4 digits for the year

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   value-format davf-yyyyymmdd  
   .
```

Decoration-Background-Visible

When this style is set, the default, column and row headings of the calendar have a visible background, otherwise the background color is not drawn.

Example - Define a date-entry control with decoration background

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   decoration-background-visible.
```

Decoration-Borders-Visible

When this style is set, a border is drawn around each item of the column and row headings of the calendar, otherwise, headings appear flat.

Example - Define a date-entry control with decoration borders

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   decoration-background-visible  
   decoration-borders-visible.
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Date-Entry control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a date-entry control with height in cells

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1  
   id 17  
   century-date  
   height-in-cells.
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a date-entry control with bold foreground

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   bold  
   century-date  
   value-format davf-yyyyymmdd  
   .
```

No-F4

Setting this style allows you to use the [F4] key as any other function key. The calendar can be opened clicking the button on the right side of the Date-Entry control. When this style is not set, which is the default, the function key [F4] opens the calendar and cannot be used as a normal function key.

Example - Define a date-entry control that does not use F4 to open up the calendar

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   no-f4  
   value-format davf-yyyyymmdd  
   .
```

No-Tab

Controls with this style set are skipped when the user navigates the screen using the Tab and the Backtab keys.

Example - Define a date-entry control that is not navigable with tab key

```
screen section.  
...  
03 screen-1-de-1 Date-Entry  
   line 21.2  
   column 49.5  
   size 14.5 cells  
   lines 3.1 cells  
   id 17  
   century-date  
   no-tab  
   value-format davf-yyyymmdd  
   .
```

No-Updown

Treated as a comment. The compiler recognizes this style for compatibility reasons.

Notify-Change

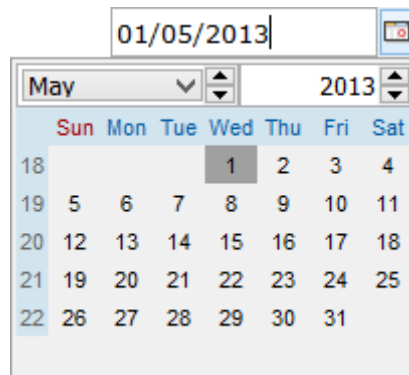
This style causes an NTF-CHANGED event to be fired each time the user changes the content of the Date-Entry control. Without this style, no event is generated under this circumstance.

Example - Define a date-entry control that fires notifies when its value changes

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   century-date  
   notify-change  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   .
```

Numeric

When this style is set, the Date-Entry layout changes. Spinner buttons are not shown anymore and the user can type numbers freely like he does in a standard Entry-Field without having to click on the year, on the month and on the day to modify their values.



This style also activates a real-time validation of the dates input by the user. The text is colored in green for valid dates and in red for invalid dates during editing.

Note - this style can't be set dynamically through MODIFY and, when this style is set, also DISPLAY-FORMAT can't be changed dynamically through MODIFY.

Example - Define a date-entry control that allows direct numeric entry

```
screen section.
...
03 screen-1-de-2 Date-Entry
   line 28.5
   column 49.7
   size 13.6 cells
   lines 4.3 cells
   id 18
   century-date
   numeric
   value-format davf-yyyyymmdd
   calendar-font Default-Font
   .
```

{ Permanent | Temporary }

Permanent

A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.

Temporary

Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a date-entry control that is temporary

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   temporary  
   .
```

Read-Only

When this style is set, the entry-field part of the control is not editable. The user can change the value only by using the spinner buttons and the calendar.

Example - Define a date-entry control that is read-only

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   read-only  
   .
```

Right-Align

This style affects the position of the date within the Date-Entry control. When set, the date is right aligned. If not set, which is the default, it is left aligned.

Example - Define a date-entry control that shows the date right aligned

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   right-align  
   .
```

Self-Act

When this style is set, all events that the Date-Entry control fires are trapped and no Event Procedure is started.

Example - Define a date-entry control that fires no events

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   self-act  
   .
```

Show-None

When this style is set, a check-box is shown on the left of the Date-Entry control.

The **Value** of the Date-Entry is set as follows according to the check-box state:

- 0 if the check-box is not checked
- 9 if the check-box is not checked and the **Time** style is set
- the control content if the check-box is checked.

Example - Define a date-entry control with the show-none style

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   century-date  
   value-format davf-yyyymmdd  
   calendar-font Default-Font  
   show-none  
   .
```

Spinner

Treated as a comment. The compiler recognizes this style for compatibility reasons.

Time

When this style is set, the Date-Entry control represents a time instead of a date. The value of the [Value-Format](#) property must be set accordingly.

Example - Define a date-entry control to handle time data

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   time.
```

Week-Of-Year-Visible

When this style is set, the default, the week of the year is shown on the left side of the calendar.

Example - Define a date-entry control that shows the week of the year on the calendar area

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6 cells  
   lines 4.3 cells  
   id 18  
   week-of-year-visible.
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Date-Entry control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a date-entry control that shows the week of the year on the calendar area

```
screen section.  
...  
03 screen-1-de-2 Date-Entry  
   line 28.5  
   column 49.7  
   size 13.6  
   lines 4.3 cells  
   id 18  
   width-in-cells.
```

Events

The following events are applicable to the DATE-ENTRY control: [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#), [NTF-CHANGED](#).

CMD-GOTO

This event is fired when the user tries to activate the Date-Entry control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Date-Entry control is requested. The EVENT-DATA-2 data item contains the Date-Entry control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when he closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

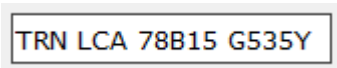
MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

NTF-CHANGED

This event is fired when the user modifies the content of an Date-Entry with the Notify-Change style set.

ENTRY-FIELD



An Entry-Field is an area where the user can enter text. It can be fully customized to meet the programmer's needs.

Properties

The following properties are applicable to the ENTRY-FIELD control: [Action](#), [Auto-Decimal](#), [Background-Color](#), [Bitmap-Disabled](#), [Bitmap-Handle](#), [Bitmap-Hint](#), [Bitmap-Number](#), [Bitmap-Rollover](#), [Bitmap-Trailing-Disabled](#), [Bitmap-Trailing-Hint](#), [Bitmap-Trailing-Number](#), [Bitmap-Trailing-Rollover](#), [Bitmap-Width](#), [Border-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Cursor](#), [Cursor-Col](#), [Cursor-Row](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Fill-Char](#), [Font](#), [Foreground-Color](#), [Format-String](#), [Format-Type](#), [Help-Id](#), [Hint](#), [Id](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Lines](#), [Max-Text](#), [Max-Val](#), [Max-Width](#), [Min-Height](#), [Min-Val](#), [Min-Width](#), [Notify-Change-Delay](#), [Pop-Up Menu](#), [Placeholder](#), [Pos](#), [Position](#), [Proposal](#), [Proposal-Delay](#), [Proposal-Index](#), [Proposal-Min-Text](#), [Proposal-To-Delete](#), [Reset-Proposals](#), [Selection-Start](#), [Selection-Start-Col](#), [Selection-Start-Row](#), [Selection-Text](#), [Size](#), [Spell-Checking](#), [Text-Orientation](#), [Validation-Errmsg](#), [Validation-Opts](#), [Validation-Regexp](#), [Value](#), [Visible](#), [Visible-Proposal-Count](#).

Action

A specific action is performed when a value is assigned to this property. The following symbolic values, included in the copy file [isgui.def](#), can be assigned. The table below shows the actions related to each value:

Action-Cut	Cuts the selected text in the entry-field to the clipboard
Action-Copy	Copies the selected text in the entry-field to the clipboard
Action-Paste	Pastes the content of the clipboard into the entry-field
Action-Delete	Deletes the currently selected text
Action-Undo	Undoes the last action
Action-Redo	Redoes the last action
Action-Select-All	Selects all the text in the control

Example - Use the action property to select the value of an entry-field and copy it to the clipboard

```
working-storage section.  
copy "isgui.def".  
...  
screen section.  
...  
03 screen-1-ef-1 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   3-d  
   .  
...  
procedure division.  
...  
   modify screen-1-ef-1 action action-select-all  
   modify screen-1-ef-1 action action-copy  
...  

```

Auto-Decimal

When set to a value greater than zero, the cursor automatically jumps to the next control as soon as the user has entered a number of decimal digits equal to the value of this property.

Example - Define an entry-field control with 2 digits of autodecimal

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   3-d  
   auto-decimal 2  
   .  

```

Background-Color

This property allows you to set or retrieve the background color of the Entry-Field control. See "[Color management](#)" for further details.

Example - Define an entry-field control with black background and gray foreground

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   background-color 0  
   foreground-color 7  
   id 19  
   3-d  
   .
```

Bitmap-Disabled

This property identifies the image to be displayed before the text when the Entry-Field control is disabled. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used for the Entry-Field.

Bitmaps are not shown in Entry-Fields with one of the following styles: [Auto-Spin](#), [Multiline](#), [Spinner](#).

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the bitmap shown before the text in the Entry-Field control.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Number

This property identifies the image to be displayed before the text when the Entry-Field control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Rollover

This property identifies the image to be displayed before the text in the Entry-Field control when the mouse pointer is moved over that area. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the

mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Trailing-Disabled

This property identifies the image to be displayed after the text when the Entry-Field control is disabled. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```


Bitmap-Trailing-Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the bitmap shown after the text in the Entry-Field control.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Trailing-Number

This property identifies the image to be displayed after the text when the Entry-Field control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the

mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Trailing-Rollover

This property identifies the image to be displayed after the text in the Entry-Field control when the mouse pointer is moved over that area. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define an entry-field control with leading and trailing images and hints do be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Bitmap-Width

This property defines the width in pixels of the image used.

Example - Define an entry-field control with leading and trailing images and hints to be displayed when the mouse goes over them

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   bitmap-handle bmp-strip  
   bitmap-width 16  
   bitmap-number 1  
   bitmap-disabled 2  
   bitmap-rollover 3  
   bitmap-hint "This is the bitmap on the left"  
   bitmap-trailing-number 4  
   bitmap-trailing-disabled 5  
   bitmap-trailing-rollover 6  
   bitmap-trailing-hint "This is the bitmap on the right"  
   id 19  
   3-d  
   .
```

Border-Color

This property allows you to set or retrieve the border color of the Entry-Field control. See "[Color management](#)" for further details. The border color is applicable only to controls with the [Boxed](#) style.

Example - Define an entry-field control with red border

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   border-color 5  
   id 19  
   3-d  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Entry-Field control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Entry-Field control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Entry-Field control will be relative to the ending position of

the prior Screen Section item.

When the Entry-Field control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).
03 Entry-Field, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a entry-field at column 5.0 on the screen section definition screen section.

```
screen section.
...
03 screen-1-ef-2 Entry-Field
   line 8.0
   column 5.0
   color 7
   size 45.0 cells
   id 2
   .
```

Color

This property allows you to set or retrieve the color of the Entry-Field control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define an entry-field control with dark gray background and gray foreground using the color property

```
screen section.
...
03 screen-1-ef-2 Entry-Field
   line 35.5
   column 49.7
   size 13.1 cells
   lines 4.6 cells
   color 296
   id 19
   3-d
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define an entry-field with css base style name, valid with EIS WD2

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   css-base-style-name "css-ef-style"  
   3-d  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define an entry-field with css style name, valid with EIS WD2

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   css-style-name "css-ef-style"  
   3-d  
   .
```

Cursor

This property set or retrieves the cursor position inside an entry-field. If the value "-1" is assigned to this property, the whole text in the entry-field is selected and the cursor is positioned at the end of it.

Note that if it is necessary to position the cursor in a multi line entry-field, the [Cursor-Col](#) and [Cursor-Row](#) properties should be used.

Example - Get the cursor position of an entry-field control

```
working-storage section.  
77 ws-cur-col pic 9(3).  
...  
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 35.5  
   column 49.7  
   size 13.1 cells  
   lines 4.6 cells  
   color 296  
   id 19  
   3-d  
   .  
...  
procedure division.  
...  
   inquire screen-1-ef-2 cursor ws-cur-col  
...
```

Cursor-Col

This property sets or retrieves the horizontal cursor position inside a multi line entry-field.

When used to set the cursor position, it must be set in conjunction with [Cursor-Row](#).

Example - Get the column and row position of the cursor on a multiline entry-field control

```
working-storage section.  
77 ws-cur-col pic 9(3).  
77 ws-cur-row pic 9(3).  
...  
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   multiline  
   3-d  
   .  
...  
procedure division.  
...  
   inquire screen-1-ef-4 cursor-col ws-cur-col cursor-row ws-cur-row  
...
```

Cursor-Row

This property sets or retrieves the vertical cursor position inside a multi line entry-field.

When used to set the cursor position, it must be set in conjunction with [Cursor-Col](#).

Example - Get the column and row position of the cursor on a multiline entry-field control

```
working-storage section.  
77 ws-cur-col    pic 9(3).  
77 ws-cur-row    pic 9(3).  
...  
screen section.  
...  
  03 screen-1-ef-4 Entry-Field  
    line 40.6  
    column 47.5  
    size 16.2 cells  
    lines 12.3 cells  
    id 21  
    multiline  
    3-d  
    .  
...  
procedure division.  
...  
    inquire screen-1-ef-4 cursor-col ws-cur-col cursor-row ws-cur-row  
...  

```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a entry-field

```
procedure division.  
...  
    modify screen-1-ef-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Entry-Field control is disabled, "1" if it is enabled.

Example - Define an entry-field control, initially disabled and enable it on procedure division

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   enabled 0  
   id 20  
   3-d  
   .  
...  
procedure division.  
...  
   modify screen-1-ef-3 enabled 1  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define an entry-field with a list of events to be excluded

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   event-list ( msg-validate ntf-changed)  
   exclude-event-list 1  
   3-d  
   .  

```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define an entry-field with a list of events to be excluded

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   event-list ( msg-validate ntf-changed)  
   exclude-event-list 1  
   3-d  
   .
```

Fill-Char

This property is used to specify what character is used to fill the empty places of an entry-field formatted with the [Format-String](#) property.

Example - Define an entry-field control with a fill char of underscores and a format-string

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   3-d  
   fill-char "_"  
   format-string "###-????-###"  
   .
```

Font

This property specifies the font used to display the content of the Entry-Field control. It may be used to compute the height and the width of the Entry-Field control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define an entry-field control with Verdana font

```
working-storage section.  
copy "isfonts.def".  
77 Verdana-10v0 handle of font.  
...  
screen section.  
...  
    03 screen-1-ef-2 Entry-Field  
        line 29.0  
        column 49.5  
        size 13.1 cells  
        lines 4.6 cells  
        font Verdana-10v0  
        id 19  
        3-d  
        .  
...  
procedure division.  
...  
    initialize wfont-data verdana-10v0.  
    move 10 to wfont-size.  
    move "Verdana" to wfont-name.  
    set wfont-bold to false.  
    set wfont-italic to false.  
    set wfont-underline to false.  
    set wfont-strikeout to false.  
    set wfont-fixed-pitch to false.  
    call "w$font" using wfont-get-font verdana-10v0 wfont-data.  
...  

```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Entry-Field control. See "[Color management](#)" for further details.

Example - Define an entry-field control with black background and gray foreground

```
screen section.  
...  
    03 screen-1-ef-2 Entry-Field  
        line 35.5  
        column 49.7  
        size 13.1 cells  
        lines 4.6 cells  
        background-color 0  
        foreground-color 7  
        id 19  
        3-d  
        .  

```

Format-String

This property is used in conjunction with the [Fill-Char](#) property to assign a special format to an entry-field. The entry-field value is immediately formatted while the user is typing.

The following special characters can be used to define the format:

#	A digit.
U	Any alphabetic character, no numbers are allowed. Letters are converted to upper-case.
L	Any alphabetic character, no numbers are allowed. Letters are converted to lower-case.
A	Any alphanumeric character.
?	Any alphabetic character, no numbers are allowed.
*	Any character, including symbols.
H	Any hex digit, namely 0-9, A-F and a-f.
Others	Characters not listed above are shown in the entry-field and are used as separators that cannot be modified.
	To be used as separators, characters described above must be preceded by a single quote (').

All characters described above can be combined to define complex formatting strings.

"###-????-###" would define a string that must be three digits, five letters and three digits, separated by dashes.

Note that the value of the field for the program will be stripped of separator characters. For example, having the format string ##/##/##, the user types "101217" and sees "10/12/17", but the program will receive only "101217".

When this property is set, LOWER, NUMERIC and UPPER styles are ignored. Rely on the Format-String syntax in order to obtain the same effect.

Example - Define an entry-field control with a fill char of underscores and a format-string

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
  line 29.0  
  column 49.5  
  size 13.1 cells  
  lines 4.6 cells  
  id 19  
  3-d  
  fill-char '_'  
  format-string "###-????-###"  
  .
```

Format-Type

This property allows you to apply a normalization to the value of the Entry-Field before returning it to the program. This property can be set to "DATE" or "NUMERIC". In the first case, the value will be formatted using java.text.SimpleDateFormat, in the second case it will be formatted using java.text.DecimalFormat.

Note: This property is deprecated and should not be used.

Help-Id

This property allows you to assign a unique ID to the Entry-Field control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define an entry-field with a Help-id

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   help-id 3090  
   id 19  
3-d.
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Entry-Field control.

Example - Define an entry-field control with a hint value

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   hint "Enter your original code here"  
3-d.
```

Id

This property allows you to assign a unique ID to the Entry-Field control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define an entry-field with an ID property

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
3-d.
```

Layout-data

The The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define an entry-field that allows resize in X and Y when the layout manager requires so

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   layout-data 17  
3-d.
```

Line

This property allows you to specify the Entry-Field control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Entry-Field control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Entry-Field control will be relative to the starting position of the prior Screen Section item.

When the Entry-Field control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Entry-Field, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position an entry-field at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Entry-Field control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Entry-Field control is still computed in CELLS, but the cell size is based on the font set for the Entry-Field control with the [Font](#) property. If no font has been defined for the Entry-Field control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define an entry-field control defining its dimensions in lines and size

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   3-d.
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define an entry-field control with max-height, min-height, max-width, min-width as limits to the

resizing

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 10.0  
   layout-data 17  
   3-d.
```

Max-Lines

The value set to this property establishes the maximum number of lines the user can enter in a multi-line entry-field.

Example - Define an entry-field control with a multiline style, limiting the lines to 4

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   multiline  
   3-d  
   max-lines 4  
   .
```

Max-Text

This property can be used to set the maximum number of characters the user can enter. If this property is not specified, it is set to the same value as the [Size](#) property.

Example - Define an entry-field control that accepts 10 characters maximum

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   max-text 10  
   .
```

Max-Val

With this property it is possible to set the maximum numeric value that can be entered in the entry-field. If the number entered is greater than the one specified here, an error message is shown.

Negative values are allowed.

Example - Define a numeric entry-field control with a max value of 999

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   numeric  
   max-val 999  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define an entry-field control with max-height, min-height, max-width, min-width as limits to the

resizing

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 10.0  
   layout-data 17  
   3-d.
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define an entry-field control with max-height, min-height, max-width, min-width as limits to the resizing

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 10.0  
   layout-data 17  
   3-d.
```

Min-Val

With this property it is possible to set the minimum numeric value that can be entered in the entry-field. If the number entered is lower than the one specified here, an error message is shown.

Negative values are allowed.

Example - Define a numeric entry-field control with a minimum and maximum value

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   numeric  
   max-val 999  
   min-val 100  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define an entry-field control with max-height, min-height, max-width, min-width as limits to the resizing

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 10.0  
   layout-data 17  
   3-d.
```

Notify-Change-Delay

This property specifies how many milliseconds the runtime has to wait before firing a [NTF-CHANGED](#) event when the user changes the content of the Entry-Field. In this way it's possible to catch multiple changes in a single [NTF-CHANGED](#) event, increasing performance.

By default, the delay specified by the `iscobol.gui.entryfield.notify_change_delay` * configuration property is used. If `iscobol.gui.entryfield.notify_change_delay` * is not set or is set to 0, then an [NTF-CHANGED](#) event is fired for each single change made by the user.

Example - Define an entry-field control with a delay of half second for change notifications

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
   notify-change  
   notify-change-delay 500  
   3-d.
```

Placeholder

This property specifies a short hint that describes the expected value of an input field. The short hint is displayed in the input field before the user enters a value.

Note - The placeholder text is not necessarily displayed in the same position as the input text.

Example - Define an entry field with a placeholder text

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   id 20  
   3-d  
   placeholder "Enter your name here"  
   .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Entry-Field control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define an entry-field with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   pop-up menu hmenu  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposal-delay 400  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Proposal

This property adds a new item to the list of proposals associated with the Entry-Field. Multiple values are allowed if specified between parenthesis. If the [Proposals-Unsorted](#) style is set, the position of the new proposal can be controlled by the [Proposal-Index](#) property.

Proposals are not supported on fields with one or more of the following attributes:

- a [Format-String](#),
- the [Auto-Decimal](#) property set to a value greater than 0,
- the { [Auto-Spin](#) | [Spinner](#) } styles,
- the [Multiline](#) style,
- the [Read-Only](#) style,
- the [Secure](#) style.

Example - Define a list of proposals for an entry-field on procedure division.

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposal-delay 400  
   .  
...  
procedure division.  
...  
  modify screen-1-ef-3  
    proposal "rome"  
    proposal "romania"  
    proposal "romantic"  
    proposal "ron"  
    proposal "roll"  
    .  
...  

```

Proposal-Delay

This Property specifies a delay expressed in milliseconds that the list of proposals will wait before popping up.

Example - Define an entry-field with a specific proposal delay

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposal-delay 400  
   .  
...  
procedure division.  
...  
  modify screen-1-ef-3  
    proposal "rome"  
    proposal "romania"  
    proposal "romantic"  
    proposal "ron"  
    proposal "roll"  
    .  
...  

```

Proposal-Index

This property can be set before [Proposal](#) in order to specify a particular position for the newly added proposal. Each proposal is identified by a number that matches its position in the list, starting from 1. The new proposal will be added right before the one specified by this property.

Example - Add one proposal text on a specific index

```
screen section.  
...  
  03 screen-1-ef-3 Entry-Field  
    line 35.8  
    column 49.7  
    size 11.6 cells  
    lines 3.5 cells  
    id 20  
    3-d  
    proposals-unsorted  
    proposal-delay 400  
    .  
...  
procedure division.  
...  
  modify screen-1-ef-3 proposal-index 3 proposal "rox"  
...  

```

Proposal-Min-Text

This property specifies how many characters must be typed by the user before the proposal list is shown. If not set, 1 is assumed.

Example - Define an entry-field where proposals are shown after the user types three characters

```
screen section.  
...  
  03 screen-1-ef-3 Entry-Field  
    line 35.8  
    column 49.7  
    size 11.6 cells  
    lines 3.5 cells  
    id 20  
    3-d  
    proposal-min-text 3  
    .  
...  
procedure division.  
...  
  modify screen-1-ef-3  
    proposal "rome"  
    proposal "romania"  
    proposal "romantic"  
    proposal "ron"  
    proposal "roll"  
    .  
...  

```

Proposal-To-Delete

This property removes a proposal from the list of proposals associated with the Entry-Field. Setting it to a value greater than zero removes the corresponding proposal from the list. Each proposal is identified by a number that matches its position in the list, starting from 1.

Example - Delete a specific proposal item from an entry-field

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposal-delay 400  
   .  
...  
procedure division.  
...  
   modify screen-1-ef-3  
       proposal "rome"  
       proposal "romania"  
       proposal "romantic"  
       proposal "ron"  
       proposal "roll"  
...  
   modify screen-1-ef-3 proposal-to-delete 3 | would delete romantic  
...
```

Reset-Proposals

When set to a non-zero value, this property removes all the proposals associated with the Entry-Field.

Example - Reset the proposals list for an entry-field

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposal-delay 400  
   .  
...  
procedure division.  
...  
   *> Reset the proposals list before adding new ones  
   modify screen-1-ef-3 reset-proposals 1  
   modify screen-1-ef-3  
       proposal "rome"  
       proposal "romania"  
       proposal "romantic"  
       proposal "ron"  
       proposal "roll"  
...  
...
```


Selection-Start

This property returns the start position of the selection of text in the Entry-Field.

Example - Get the selection start position from an entry-field control

```
working-storage section.
...
77 ws-selection-start  pic 9(3).
...
screen section.
...
03 screen-1-ef-3  Entry-Field
   line 35.8
   column 49.7
   size 11.6 cells
   lines 3.5 cells
   id 20
   3-d
   .
...
procedure division.
...
   inquire screen-1-ef-3  selection-start ws-selection-start
...

```

Selection-Start-Col

This property returns the X start position of the selection of text in a [Multiline](#) Entry-Field.

Example - Get the selection start column and row from a multiline entry-field control

```
working-storage section.
...
77 ws-sel-start-col  pic 9(3).
77 ws-sel-start-row  pic 9(3).
...
screen section.
...
03 screen-1-ef-4  Entry-Field
   line 40.6
   column 47.5
   size 16.2 cells
   lines 12.3 cells
   id 21
   multiline
   3-d
   max-lines 4.
...
procedure division.
...
   inquire screen-1-ef-4  selection-start-col ws-sel-start-col
                           selection-start-row ws-sel-start-row
...

```

Selection-Start-Row

This property returns the Y start position of the selection of text in a [Multiline](#) Entry-Field.

Example - Get the selection start column and row from a multiline entry-field control

```
working-storage section.
...
77 ws-sel-start-col pic 9(3).
77 ws-sel-start-row pic 9(3).
...

screen section.
...
03 screen-1-ef-4 Entry-Field
   line 40.6
   column 47.5
   size 16.2 cells
   lines 12.3 cells
   id 21
   multiline
   3-d
   max-lines 4.
...
procedure division.
...
   inquire screen-1-ef-4 selection-start-col ws-sel-start-col
                        selection-start-row ws-sel-start-row
...

```

Selection-Text

With this property it is possible to substitute the currently selected text in the entry-field with the one assigned to this property.

Example - Replace the selection text by another text

```
screen section.
...
03 screen-1-ef-4 Entry-Field
   line 40.6
   column 47.5
   size 16.2 cells
   lines 12.3 cells
   id 21
   3-d
   .
...
procedure division.
...
   modify screen-1-ef-4 selection-text "new text"
...

```

Size

This property allows you to specify the size of the Entry-Field control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Entry-Field control is still computed in CELLS, but the cell size is based on the font set for the Entry-Field control with the [Font](#) property. If no font has been defined for the Entry-Field control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define an entry-field control defining its dimensions in lines and size

```
screen section.  
...  
03 screen-1-ef-2 Entry-Field  
   line 29.0  
   column 49.5  
   size 13.1 cells  
   lines 4.6 cells  
   id 19  
3-d.
```

Spell-Checking

This property allows you to enable a real time spell checking on the field.

Setting this property to "default" activates the spell checking using the current language defined in the environment.

Setting this property to a specific Locale ID activates the spell checking using a specific language of a specific country. The supported Locale IDs are: ast-ES, be-BY, br-FR, ca, ca-ES, ca-ES-valencia, da-DK, de, de-AT, de-CH, de-DE, de-DE-x-simple-language, el-GR, en, en-AU, en-CA, en-GB, en-NZ, en-US, en-ZA, eo, es, fa, fr, gl-ES, is-IS, it, ja-JP, km-KH, lt-LT, ml-IN, nl, pl-PL, pt, pt-BR, pt-PT, ro-RO, ru-RU, sk-SK, sl-SI, sv, ta-IN, tl-PH, uk-UA, xx-XX, zh-CN. For more information about these values, refer to the table available in the Java documentation: <http://www.oracle.com/technetwork/java/javase/javase7locales-334809.html>.

The mistyped words are underlined and leaving the mouse pointer over them spawns a tool-tip that suggests the proper correction. The tool-tip delay is controlled by the configuration property [iscobol.gui.entryfield.spell_checking_delay](#) whose default is a half second.

Setting this property to spaces disables the feature.

At run time, the spell checking feature requires additional items that need to be installed separately.

1. Download one of the the following files depending on the Java version associated to your isCOBOL SDK:

Java 8 or previous	Java 9 or greater
https://www.languagetool.org/download/LanguageTool-3.1.zip	https://www.languagetool.org/download/LanguageTool-4.3.zip

2. Unzip the file in a folder of your choice. The following structure will be created:

```
CHANGES.md
CHANGES.txt
COPYING.txt
languagetool-commandline.jar
languagetool-server.jar
languagetool.jar
libs
META-INF
org
README.md
testrules.bat
testrules.sh
third-party-licenses
```

3. Don't alter the folder structure and add the full path of *languagetool.jar* to the Classpath. For example, if you unzipped the file to the folder *C:\LanguageTool* on Windows, use the following command:

```
set CLASSPATH=C:\LanguageTool\languagetool.jar;%CLASSPATH%
```

If you wish to alter the folder structure, then you should add all these items to the Classpath separately:

- o all the jars that were in the root folder
- o all the jars that were in the "lib" subfolder
- o the folder containing the "org" subfolder

In a thin client installation the Language Tool items must be installed client side and must be available in the client side Classpath.

Example - Define an entry-field control with spell checking using the default locale:

```
screen section.
...
03 screen-1-ef-3 Entry-Field
   line 35.8
   column 49.7
   size 11.6 cells
   lines 3.5 cells
   id 20
   3-d
   spell-checking "default"
   .
```

Text-Orientation

This property specifies the text orientation adopted while the user is typing text into the field. Possible values are:

0 (or omitted)	default
1	left to right

2	right to left
---	---------------

Example - Define an entry-field control with right-to-left text orientation

```
screen section.
...
03 screen-1-ef-3 Entry-Field
   line 35.8
   column 49.7
   size 11.6 cells
   lines 3.5 cells
   id 20
   3-d
   text-orientation 2
   .
```

Validation-Errmsg

This property specifies the text of the message shown by the runtime when the Entry-Field content doesn't match with the regular expression specified by [Validation-Regexp](#).

Example - Define an entry-field that accepts digits only, validating with regular expression and defining particular error message when the value is incorrect

```
screen section.
...
03 screen-1-ef-3 Entry-Field
   line 35.8
   column 49.7
   size 11.6 cells
   lines 3.5 cells
   id 20
   3-d
   validation-regexp "[0-9]*"
   validation-errmsg "Please enter a valid value: digits only"
   .
```

Validation-Opts

This numeric property specifies how the Entry-Field content will be compared with the regular expression specified by [Validation-Regexp](#). The value of this property is the sum between one or more of the following values:

Value	Description
1	case insensitive
2	left trimmed
4	right trimmed

For example, if you want a case insensitive comparison with the value right trimmed, set this property to 5.

Example - Define an entry-field control that allows digits only, but the left side may contain spaces

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   validation-regexp "[0-9]*"  
   validation-errmsg "Please enter a valid value: digits only"  
   validation-opts 2  
   .
```

Validation-Regexp

This property allows you to specify a regular expression for the validation of the Entry-Field content. When the user leave the field, if the content doesn't match with the regular expression, then a message box is shown and the focus is kept on the field. The message box text is controlled by the property [Validation-ErrMsg](#). The validation is performed according to the settings specified by [Validation-Opts](#).

Example - Define an entry-field that accepts digits only, validating with regular expression and defining particular error message when the value is incorrect

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   validation-regexp "[0-9]*"  
   validation-errmsg "Please enter a valid value: digits only"  
   .
```

Value

This property represents the value of the Entry-Field control.

When inquired, it returns the value that is currently represented.

When set, the Entry-Field control changes its look to represent it.

You may use the VALUE IS MULTIPLE option with [Multiline](#) entry fields. The value data item should be a one-dimensional OCCURS with no subscript specified. The effect of the MULTIPLE phrase is to match each line of the entry field to occurrences in the OCCURS. The first line is matched to the first occurrence, the second line with the second occurrence, and so on. Occurrences that are larger than the number of lines in the entry field are set to spaces when the entry field is accepted. If the [Max-Lines](#) property is omitted, it's automatically set to the capacity of the associated OCCURS.

Example - Define an entry-field with initial value

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   value "Initial value"  
   .
```

Visible

This property assumes a value of "0" if the Entry-Field control is not visible, "1" if it is visible.

Example - Define an entry-field initially invisible to make it visible on produce division later

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   visible 0  
   .  
...  
procedure division.  
...  
   modify screen-1-ef-3 visible 1  
...  
...
```

Visible-Proposal-Count

This property specifies how many items must be visible in the list of proposals and in consequence the height of that list. If the number of matching proposals is greater than the value of this Property, then scroll-bars are shown.

In WD2 environment, scroll-bars are not shown, so setting this property causes a truncation of the list of available entries.

Example - Define an entry-field control with a proposal list and limit the visible list count

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposal-delay 400  
   visible-proposal-count 3  
   .  
...  
procedure division.  
...  
   modify screen-1-ef-3  
       proposal "rome"  
       proposal "romania"  
       proposal "romantic"  
       proposal "ron"  
       proposal "roll"  
       .  
...  
...
```

Styles

The following styles are applicable to the ENTRY-FIELD control: 3-D, Auto, Auto-Spin, Background-High, Background-Low, Background-Standard, Bold, Boxed, Center, Centered, Height-In-Cells, High, Highlight, Left, Low, Lower, Lowlight, Multiline, No-Autosel, No-Box, No-Tab, Notify-Change, Numeric, Permanent, Proposals-Unsorted, Read-Only, Required, Right, Secure, Spinner, Standard, Temporary, Upper, Use-Return, Use-Tab, Vscroll, Vscroll-Bar, Width-In-Cells.

{ 3-D | Boxed | No-Box }

3-D	The box drawn around the Entry-Field control appears with a 3-D effect.
Boxed	A flat box is drawn around the Entry-Field control.
No-Box	No box is drawn around the Entry-Field control. Set this style when you need to save space.

The visual result may vary with different Swing LAF (Look And Feel).

Example - Define a boxed entry-field

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   boxed  
   .
```

{ Auto | Auto-Skip | Autoterminate }

Auto	This style causes the entry field to terminate as soon as it is filled by the user. A field is considered filled when the number of characters it contains equals its Max-Text setting.
------	---

Example - Define an entry-field that auto skips when it is full

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   auto  
   .
```

{ Auto-Spin | Spinner }

Auto-Spin	This style is equivalent to the SPINNER style, but here the spinner handling is simplified. When the user clicks the up or the down arrow, the entry-field value is automatically increased or decreased by 1. The valid range of values is controlled by the Min-Val and Max-Val properties.
Spinner	If this style is applied, the spinner arrows are displayed on the left of the entry-field to increment and decrement the entry-field content. The MSG-SPIN-UP event and the MSG-SPIN-DOWN event are fired when the user clicks the up or the down arrow. It is programmer's responsibility to activate procedures to increase or decrease the entry-field value.

Example - Define an entry-field with auto-spin style

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   auto-spin  
   .
```

Example - Define an entry-field with the spinner style and program the events of spin up and down to add 5 and subtract 5 to the current value respectively

```
working-storage section.
77 ws-num    pic 9(4) .
...
screen section.
...
03 screen-1-ef-5 Entry-Field
   line 50.1
   column 19.9
   size 10.7 cells
   lines 3.0 cells
   id 23
   spinner
   event procedure screen-1-ef-5-evt-proc
   3-d
   value 0
   .
...
procedure division.
...
screen-1-ef-5-evt-proc.
   evaluate event-control-id
   when 23
      evaluate event-type
      when msg-spin-up    perform screen-1-ef-5-evt-msg-spin-up
      when msg-spin-down perform screen-1-ef-5-evt-msg-spin-down
      when other
         end-evaluate
      end-evaluate.

screen-1-ef-5-evt-msg-spin-up.
   inquire screen-1-ef-5 value ws-num
   add 5 to ws-num
   modify screen-1-ef-5 value ws-num
   .
screen-1-ef-5-evt-msg-spin-down.
   inquire screen-1-ef-5 value ws-num
   if ws-num > 0
      add -5 to ws-num
      modify screen-1-ef-5 value ws-num
   end-if
   .
...
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define an entry-field control with background-high

```
screen section.  
...  
03 screen-1-ef-5 Entry-Field  
   line 50.1  
   column 19.9  
   size 10.7 cells  
   lines 3.0 cells  
   id 23  
   3-d  
   background-high  
   .
```

{ [Center | Centered] | Left | Right }

Center, Centered	The content of the entry-field is centered
Left	The content of the entry-field is left aligned
Right	The content of the entry-field is right aligned

Note - these styles have no effect on Entry-Fields with the [Multiline](#) style.

Example - Define an entry-field control with centered content

```
screen section.  
...  
03 screen-1-ef-5 Entry-Field  
   line 50.1  
   column 19.9  
   size 10.7 cells  
   lines 3.0 cells  
   id 23  
   3-d  
   center  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Entry-Field control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value](#) CELLS".

Example - Define an entry-field control with height in cells style

```
screen section.  
...  
03 screen-1-ef-5 Entry-Field  
   line 50.1  
   column 19.9  
   size 10.7 cells  
   lines 3.0  
   id 23  
   3-d  
   height-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define an entry-field control with bold foreground

```
screen section.  
...  
03 screen-1-ef-5 Entry-Field  
   line 50.1  
   column 19.9  
   size 10.7 cells  
   lines 3.0  
   id 23  
   3-d  
   bold  
   .
```

{ **Lower** | **Upper** }

Lower	When this style is set, any value entered in the entry-field is automatically converted to lower-case characters.
Upper	When this style is set, any value entered in the entry-field is automatically converted to upper-case characters.

If [Format-String](#) is set, these styles have no effect.

Example - Define an entry-field control with upper style to convert all entry to uppercase

```
screen section.  
...  
03 screen-1-ef-5 Entry-Field  
   line 50.1  
   column 19.9  
   size 10.7 cells  
   lines 3.0  
   id 23  
   3-d  
   upper  
   .
```

Multiline

When this style is set, the content of the entry-field cannot be scrolled horizontally anymore and is automatically span to multiple lines.

When the [Lines](#) property is set to a value greater than 2, this style is implied, unless the CELLS clause is specified or the [Height-In-Cells](#) style is set.

The [Value](#) of a multiline entry-field is returned as a single string obtained from the concatenation of the multiple lines separated by the character specified by the configuration property [iscobol.gui.ef_lineseparator](#).

Example - Define an entry-field control with multiline style

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   multiline  
   3-d  
   max-lines 4.
```

No-Autosel

When this style is set, the content of the entry-field is not automatically selected when activated.

Example - Define an entry-field control that does not autoselect contents when visited

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   no-autosel  
   3-d  
   .
```

No-Tab

Controls with this style set are skipped when the user navigates the screen using the Tab and the Backtab keys.

Example - Define an entry-field control that cannot be visited when pressing the tab key

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   no-tab  
   3-d  
   .
```

No-Wrap

This style causes [Multiline](#) entry-fields to show a horizontal scroll-bar when the text of a line goes over the field size. Without this style, the exceeding text is automatically moved to the next line and no scroll-bars are shown.

Example - Define an entry-field control that doesn't wrap text

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   multiline  
   no-wrap  
   3-d  
   .
```

Notify-Change

This style causes a [NTF-CHANGED](#) event to be fired each time the user changes the content of the Entry-Field control. Without this style, no event is generated under this circumstance.

Example - Define an entry-field control that fires the ntf-changed event when its value changes

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   notify-change  
   3-d  
   .
```

Numeric

This style allows users to enter only numeric data. However, entry-fields allow you to automatically enter only numeric data if a numeric or a numeric-edited data item has been assigned to the [Value](#) property. For this reason, this style is useful if only numeric data have to be accepted and the [Value](#) property is set to a non-numeric data item.

If [Format-String](#) is set, this style has no effect.

Example - Define an entry-field control that allows numeric entry only

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   numeric  
   3-d  
   .
```


{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define an entry-field control with temporary style

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   temporary  
   3-d  
   .
```

Proposals-Unsorted

When this style is set, the values of the proposals list are shown in the order they were added by the program. Without this style, proposals are shown in alphabetical order.

Example - Define an entry-field that keeps the list of proposal unsorted

```
screen section.  
...  
03 screen-1-ef-3 Entry-Field  
   line 35.8  
   column 49.7  
   size 11.6 cells  
   lines 3.5 cells  
   id 20  
   3-d  
   proposals-unsorted  
   proposal-delay 400  
   .
```

Read-Only

When this style is set, the user cannot change the content of the Entry-Field control. However, he can scroll, select and copy it.

The configuration properties `iscobol.gui.entryfield.read_only_color` and `iscobol.gui.entryfield.read_only_cursor_arrow` (boolean) affect color and behavior of read-only Entry-Fields.

Example - Define an entry-field that does not allow input

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   read-only  
   3-d  
   .
```

Secure

This style causes the content of the entry-field to be displayed as all "*". Normally used for passwords.

Example - Define an entry-field for password management with secured character entry and display

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   hint "Enter password here"  
   id 21  
   secure  
   3-d  
   .
```

Required

When this style is set, the user will not be able to leave the field empty.

Example - Define an entry-field control as a mandatory field

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   hint "Do not leave this field empty, please"  
   id 21  
   required  
   3-d  
   .
```

Use-Return

When this style is set, the user can type the [Enter] key to add a new line in a multiline entry-field. Without it, the [Enter] key terminates the input and a new line can be added by pressing [Ctrl+Enter].

Example - Define a multiline entry-field that allows the use of the ENTER key to separate lines

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   multiline  
   3-d  
   use-return.
```

Use-Tab

When this style is set, the user can use the [Tab] key to add a tab character in an entry-field. Without it, the [Tab] key is used to move from control to control.

Example - Define an entry-field that allows the use of the TAB key to input a tab character in the field

```
screen section.  
...  
03 screen-1-ef-4 Entry-Field  
   line 40.6  
   column 47.5  
   size 16.2 cells  
   lines 12.3 cells  
   id 21  
   multiline  
   3-d  
   use-tab.
```

{ Vscroll | Vscroll-Bar }

Vscroll	The user can scroll the content of a multi line entry-field with the up/down arrows. In this way, the height of the entry-field can be lower than the number of lines it can display.
Vscroll-Bar	Same as the VSCROLL style, but here a vertical scroll bar is displayed on the right side of the entry-field.

By setting either the VSCROLL style or the VSCROLL-BAR style, the [Multiline](#) style is implied.

Example - Define an entry-field with multiline style, that has more lines that can be visible and a vertical scroll-bar to scroll thru them

```
screen section.
...
03 screen-1-ef-4 Entry-Field
   line 40.6
   column 47.5
   size 22.2 cells
   lines 12.3 cells
   id 21
   multiline
   Vscroll-Bar
   3-d
   use-return
   use-tab
   max-lines 10
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Entry-Field control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define an entry-field with width in cells

```
screen section.
...
03 screen-1-ef-4 Entry-Field
   line 40.6
   column 47.5
   size 22.2
   lines 12.3 cells
   id 21
   3-d
   width-in-cells
   .
```

Events

The following events are applicable to the ENTRY-FIELD control: [CMD-GOTO](#), [CMD-HELP](#), [MSG-BITMAP-](#)

CLICKED, MSG-BITMAP-DBLCLICK, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-SPIN-DOWN, MSG-SPIN-UP, MSG-VALIDATE, NTF-CHANGED.

CMD-GOTO

This event is fired when the user tries to activate the Entry-Field control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Entry-Field control is requested. The EVENT-DATA-2 data item contains the [Help-Id](#) for the Entry-Field control.

MSG-BITMAP-CLICKED

This event is fired when one of the bitmaps displayed within the Entry-Field is clicked. The EVENT-DATA-1 data item tells which bitmap was clicked. A value of 1 indicates a click on the bitmap shown before the text. A value of 2 indicates a click on the bitmap shown after the text.

MSG-BITMAP-DBLCLICK

This event is fired when the user double clicks on one of the bitmaps displayed within the Entry-Field. The EVENT-DATA-1 data item tells which bitmap was clicked. A value of 1 indicates a double click on the bitmap shown before the text. A value of 2 indicates a double click on the bitmap shown after the text.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-SPIN-DOWN

This event is fired when the user clicks the down button of an entry-field with either the [AUTO-SPIN](#) or the [SPINNER](#) set. Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the field value from being decremented.

MSG-SPIN-UP

This event is fired when the user clicks the up button of an entry-field with either the [AUTO-SPIN](#) or the [SPINNER](#) set. Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the field value from being incremented.

MSG-VALIDATE

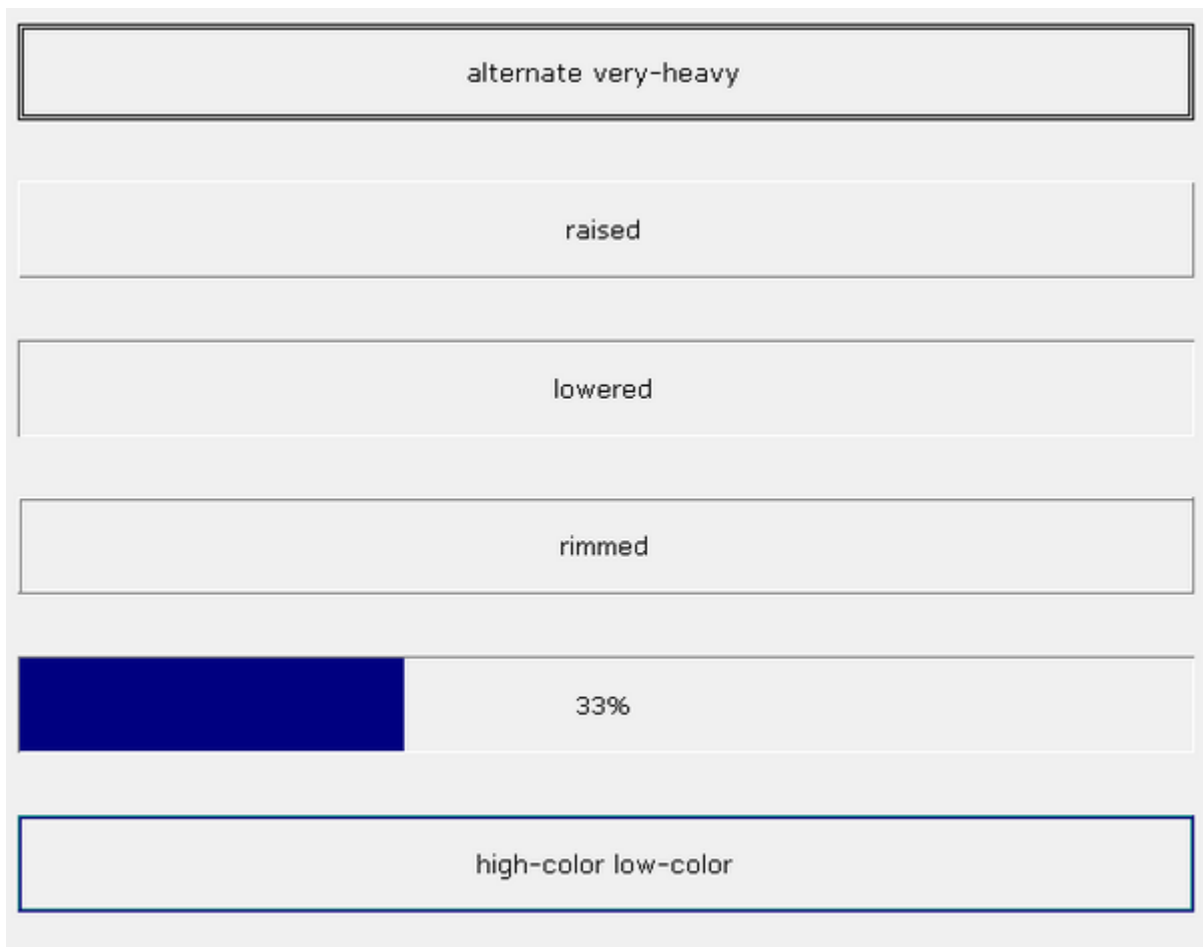
This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

NTF-CHANGED

This event is fired when the user modifies the content of an entry-field with the [Notify-Change](#) style set. The EVENT-DATA-1 data item contains the current cursor position in the entry-field.

It's possible to specify a delay for this event in order to catch more changes with a single notification. Set [Notify-Change-Delay](#) to specify the delay.

FRAME



A Frame is a box that has a merely cosmetic purpose. It can be used to visually define areas on the screen.

Properties

The following properties are applicable to the FRAME control: [Background-Color](#), [Col](#), [Color](#), [Column](#), [Css-](#)

Base-Style-Name, Css-Style-Name, Custom-Data, Fill-Color, Fill-Color2, Fill-Percent, Font, Foreground-Color, Help-Id, High-Color, Hint, Id, Layout-data, Line, Lines, Low-Color, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Title, Title-Position, Visible.

Background-Color

This property allows you to set or retrieve the background color of the Frame control. This color is applied to the background of the Frame title, if any. See "[Color management](#)" for further details.

Example - Define a frame control with background and foreground color

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   background-color 3  
   foreground-color 12  
   id 16  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Frame control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Frame control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Frame control will be relative to the ending position of the prior Screen Section item.

When the Frame control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Frame, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a frame at column 5.0 on the screen section definition screen section

```
03 screen-1-fr-1 Frame  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   lines 60.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Frame control. Foreground and background color

values are combined and therefore RGB colors are not supported. See ["Color management"](#) for further details.

In a Frame control, the foreground color is applied to the title text and to the borders while the background color is applied to the title label.

Example - Define a frame control with color property

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   color 6  
   id 16  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a frame control with a css base style, valid for EIS WD2

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   color 6  
   id 16  
   css-base-style-name "css-frame-style"  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a frame control with a css style, valid for EIS WD2

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   color 6  
   id 16  
   css-style-name "css-frame-style"  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a frame

```
procedure division.  
...  
   modify screen-1-fr-1 custom-data "Screen 1 custom data"  
...  

```

Fill-Color

This property sets the color to be used to fill the frame area. See "[Color management](#)" for further details.

Example - Define a frame control with a white fill color

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   fill-color 16  
   .
```

Fill-Color2

This property, used in conjunction with the [Fill-Percent](#) property, defines the color used to fill the portion of frame not filled with the color set in the [Fill-Color](#) property, when the value of the [Fill-Percent](#) property is less than 100. The default value, zero, indicates that no color must be applied to that frame portion. See "[Color management](#)" for further details.

Example - Define a frame control with 2 fill colors, white and gray, the fill-percent tells the percentage for the first fill color and the rest is for the other fill color

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   fill-color 16  
   fill-color2 9  
   fill-percent 73  
   .
```

Fill-Percent

This property sets the portion of the Frame control to be filled with the color set in the [Fill-Color](#) property. If the frame is wider than it is tall, it is horizontally filled starting from the left side. Otherwise, it is vertically filled from the bottom. The value of to this property must be in the range 0-100.

Example - Define a frame control with 2 fill colors, white and gray, the fill-percent tells the percentage for the first fill color and the rest is for the other fill color

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   fill-color 16  
   fill-color2 9  
   fill-percent 73  
   .
```

Font

This property specifies the font used to display the content of the Frame control. It may be used to compute the height and the width of the Frame control as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a frame control with a font to determine its size

```
working-storage section.  
copy "isfonts.def".  
77 Arial-12v0 handle of font.  
...  
screen section.  
...  
    03 screen-1-fr-1 Frame  
        line 17.7  
        column 20.0  
        size 24.3 cells  
        lines 27.8 cells  
        color 6  
        font Arial-12v0  
        id 16  
    .  
...  
procedure division.  
...  
    initialize wfont-data arial-12v0.  
    move 12 to wfont-size.  
    move "Arial" to wfont-name.  
    set wfont-bold to false.  
    set wfont-italic to false.  
    set wfont-underline to false.  
    set wfont-strikeout to false.  
    set wfont-fixed-pitch to false.  
    call "w$font" using wfont-get-font arial-12v0 wfont-data.  
...  

```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Frame control. This color is applied to the title text and to the borders. See "[Color management](#)" for further details.

Example - Define a frame control with background and foreground color

```
screen section.  
...  
    03 screen-1-fr-1 Frame  
        line 17.7  
        column 20.0  
        size 24.3 cells  
        lines 27.8 cells  
        background-color 3  
        foreground-color 12  
        id 16  
    .  

```

Help-Id

This property allows you to assign a unique ID to the Frame control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a frame control with a help-id

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   help-id 3095  
   id 16  
   .
```

High-Color

In order to obtain a 3-D effect, it is necessary to define a dark and a light color. This property defines the light color. See also the LOW-COLOR property.

See "[Color management](#)" for further details.

Example - Define a 3d raised frame control with high and low colors

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   font Arial-12v0  
   id 16  
   lowered  
   low-color rgb 4130824  
   high-color rgb 16777215  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Frame control.

Example - Define a frame control with hint text

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   hint "Customer data area"  
   id 16  
   .
```

Id

This property allows you to assign a unique ID to the Frame control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a frame control with ID property

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   .
```

Layout-data

The The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a frame control that resizes in X and Y if the layout manager requires so

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   layout-data 17  
   .
```

Line

This property allows you to specify the Frame control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Frame control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Frame control will be relative to the starting position of the prior Screen Section item.

When the Frame control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).
03 Frame, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a frame at line 8.0 on the screen section definition

```
screen section
...
03 screen-1-fr-1 Frame
   line 8.0
   column 5.0
   color 7
   size 45.0 cells
   id 2
.
```

Lines

This property allows you to specify the height of the Frame control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Frame control is still computed in CELLS, but the cell size is based on the font set for the Frame control with the [Font](#) property. If no font has been defined for the Frame control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a frame control defining its height in lines

```
screen section.
...
03 screen-1-fr-1 Frame
   line 17.7
   column 20.0
   size 24.3 cells
   lines 27.8 cells
   id 16
.
```

Low-Color

In order to obtain a 3-D effect, it is necessary to define a dark and a light color. This property defines the dark color. See also the [High-Color](#) property.

See "[Color management](#)" for further details.

Example - Define a 3d raised frame control with high and low colors

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   lowered  
   low-color rgb 4130824  
   high-color rgb 16777215  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a frame with layout-data that allows resize X and resize Y and has limits for max height, max width, min height and min width

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   max-width 50.0  
   min-width 15.0  
   min-height 20.0  
   max-height 50.0  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a frame with layout-data that allows resize X and resize Y and has limits for max height, max

width, min height and min width

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   max-width 50.0  
   min-width 15.0  
   min-height 20.0  
   max-height 50.0  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a frame with layout-data that allows resize X and resize Y and has limits for max height, max width, min height and min width

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   max-width 50.0  
   min-width 15.0  
   min-height 20.0  
   max-height 50.0  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a frame with layout-data that allows resize X and resize Y and has limits for max height, max

width, min height and min width

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   max-width 50.0  
   min-width 15.0  
   min-height 20.0  
   max-height 50.0  
   .
```

Pop-Up Menu

This property is not considered by the Frame control.

Size

This property allows you to specify the size of the Frame control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Frame control is still computed in CELLS, but the cell size is based on the font set for the Frame control with the [Font](#) property. If no font has been defined for the Frame control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a frame control with size in cells

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   font Arial-12v0  
   id 16  
   .
```

Title

The description shown in the Frame control. The [Title-Position](#) property affects the position of the text.

Example - Define a frame with a top centered title

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   title-position 2  
   title "Customer Data Area"  
   .
```

Title-Position

This property affects the position of the title. Valid values are:

1	Top left, the default.
2	Top center
3	Top right
4	Bottom left
5	Bottom center
6	Bottom right
7	Horizontally and vertically centered

Example - Define a frame with a top centered title

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   title-position 2  
   title "Customer Data Area"  
   .
```

Visible

This property assumes a value of "0" if the Frame control is not visible, "1" if it is visible.

Example - Define a frame initially invisible to make it visible later on procedure division

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   title-position 2  
   title "Customer Data Area"  
   visible 0  
   .  
...  
procedure division.  
...  
   modify screen-1-fr-1 visible 1  
...
```

Styles

The following styles are applicable to the FRAME control: [Alternate](#), [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Engraved](#), [Full-Height](#), [Heavy](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Low](#), [Lowered](#), [Lowlight](#), [Permanent](#), [Raised](#), [Rimmed](#), [Standard](#), [Temporary](#), [Transparent](#), [Very-Heavy](#), [Width-In-Cells](#).

Alternate

This style, used only in conjunction with the [VERY-HEAVY](#) style, draws a double line border.

Example - Define a frame with double border using alternate and very-heavy styles

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   very-heavy  
   alternate.
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a frame with high background

```
screen section.  
...  
03 screen-1-fr-1 Frame  
    line 17.7  
    column 20.0  
    size 24.3 cells  
    lines 27.8 cells  
    id 16  
    background-high.
```

{ Engraved | Lowered | Raised | Rimmed }

Engraved	This style causes the frame border to appear engraved with respect to the screen surface, while the interior area appears at the same level.
Lowered	This style causes the frame area to appear lowered with respect to the screen surface.
Raised	This style causes the frame area to appear raised with respect to the screen surface.
Rimmed	This style causes the frame border to appear raised with respect to the screen surface, while the interior area appears at the same level.

Example - Define a 3d raised frame control with high and low colors

```
screen section.  
...  
03 screen-1-fr-1 Frame  
    line 17.7  
    column 20.0  
    size 24.3 cells  
    lines 27.8 cells  
    id 16  
    lowered  
    low-color rgb 4130824  
    high-color rgb 16777215  
    .
```

Full-Height

When this style is not set, the upper line of the frame, where the title is usually shown, is drawn in a way that it is centered with respect to the title itself. Thus, the upper corners are not drawn at the exact position defined by the [Line](#) property, but half line below.

By setting this style, no adjustment is done and the frame starts exactly where defined.

The overall height of the frame is not affected by the setting of this style.

Example - Define a frame with full-height style

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   full-height  
   .
```

{ Heavy | Very-Heavy }

Heavy	When this style is set, the border is thicker than normal.
Very-Heavy	When this style is set, the border is even thicker. When it is set in conjunction with the Alternate style, the frame is drawn with a double border.

Example - Define a frame with double border using alternate and very-heavy styles

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8 cells  
   id 16  
   very-heavy  
   alternate.
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Frame control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines](#) value CELLS".

Example - Define a frame with height in cells

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3 cells  
   lines 27.8  
   id 16  
   height-in-cells.
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a frame with bold foreground

```
screen section.  
...  
03 screen-1-fr-1 Frame  
    line 17.7  
    column 20.0  
    size 24.3 cells  
    lines 27.8  
    id 16  
    bold.
```

{ **Permanent** | **Temporary** }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary style frame control

```
screen section.  
...  
03 screen-1-fr-1 Frame  
    line 17.7  
    column 20.0  
    size 24.3 cells  
    lines 27.8  
    id 16  
    temporary.
```

Transparent

When this style is set, the title background becomes transparent.

Example - Define a frame with transparent title background

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3  
   lines 27.8 cells  
   id 16  
   title "Options"  
   transparent.
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Frame control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a frame with height in cells

```
screen section.  
...  
03 screen-1-fr-1 Frame  
   line 17.7  
   column 20.0  
   size 24.3  
   lines 27.8 cells  
   id 16  
   width-in-cells.
```

Events

The Frame control doesn't fire events.

GRID

	Description	QT	Price	%	Date	Y/n
01	WACOM graphic tablet INTUOS3 Classic A5	16	223,58	60	01/01/2012	<input type="checkbox"/>
02	WACOM graphic tablet GRAPHIRE3	05	238,80	100	01/02/2012	<input checked="" type="checkbox"/>
03	WACOM graphic tablet GRAPHIRE3 Studio	52	123,96	90	01/03/2012	<input type="checkbox"/>
04	WACOM graphic tablet GRAPHIRE3 Classic	23	99,00	40	01/04/2012	<input checked="" type="checkbox"/>
05	WACOM graphic tablet VOLITO 2 A6 USB	50	50,00	60	01/05/2012	<input type="checkbox"/>
06	WACOM graphic tablet INTUOS3 A5 USB	45	342,36	100	01/06/2012	<input checked="" type="checkbox"/>
07	WACOM graphic tablet INTUOS3 A6	65	237,96	70	01/07/2012	<input checked="" type="checkbox"/>
08	WACOM graphic tablet INTUOS3 A4 USB	03	490,68	100	01/08/2012	<input type="checkbox"/>
09	WACOM CintiQ graphic tablet 17 TFT	15	2632,56	100	01/09/2012	<input checked="" type="checkbox"/>
10	NGS graphic tablet Draw Master	63	59,00	80	01/10/2012	<input checked="" type="checkbox"/>
11	NGS graphic tablet Cadboy 14x10cm USB	14	38,00	10	01/11/2012	<input type="checkbox"/>
12	WACOM CintiQ graphic tablet 18 TFT	00	3373,44	100	01/12/2012	<input checked="" type="checkbox"/>

A Grid is a two-dimensional table. It can be customized to meet programmer's needs and cells can hold controls, such as Combo-Boxes, Push-Buttons and Entry-Fields.

Information is organized in rows and columns, and the cells are addressed by row and column coordinates.

Colors and Fonts

Since each element of a Grid can be rendered with different colors and fonts and elements can be overlapped, priority rules are needed.

Colors are applied according to the following list of priorities. Items are listed from the most important to the least important:

- Cursor-Color, Cursor-Foreground-Color, Cursor-Background-Color
- Drag-Color, Drag-Foreground-Color, Drag-Background-Color
- Region-Color, Region-Foreground-Color, Region-Background-Color
- Cell-Color, Cell-Foreground-Color, Cell-Background-Color
- Heading-Color, Heading-Foreground-Color, Heading-Background-Color
- Row-Cursor-Color, Row-Cursor-Foreground-Color, Row-Cursor-Background-Color
- Row-Color, Row-Foreground-Color, Row-Background-Color
- Column-Color, Column-Foreground-Color, Column-Background-Color
- Row-Color-Pattern, Row-Foreground-Color-Pattern, Row-Background-Color-Pattern
- Color, Foreground-Color, Background-Color

Fonts are applied according to the following list of priorities. Items are listed from the most important to the least important:

- Cell-Font
- Heading-Font
- Row-Font


- [Column-Font](#)
- [Font](#)

Protections are applied according to the following list of priorities. Items are listed from the most important to the least important:

- [Cell-Protection](#)
- [Row-Protection](#)
- [Column-Protection](#)
- [Protection](#)

Filtering and finding data

Grids provide a integrated "find all" feature that is activated by pressing CTRL+F when the focus is on the Grid. When CTRL+F is pressed, the following panel is shown on top of the Grid:



The X button on the left allows to close the panel. When the panel is closed, the Grid data is restored.

The Combo-Box in the middle allows to input the text you're looking for. As you type text in the field, the Grid rows that don't include your text are made invisible. The other rows show the matching text highlighted unless the [Alignment](#) of the column is "H".

Pressing Enter or clicking the Find button stores your text in the Combo-Box. If you wish to search the same text later, you can select it from the Combo-Box instead of typing it from scratch. The text remain stored until the Grid is destroyed.

The text inputted is evaluated as regular expression, so the search is case sensitive by default and special characters like the dollar sign will not match unless they're escaped. If the text inputted is not a valid regular expression, then the field border gets colored in red.

Clicking the Clear button clears the text area of the Combo-Box and restores all the data in the Grid.

The integrated "find all" feature can be disabled by applying the [No-Search](#) style.

The [Filterable-Columns](#) style instead allows to filter data according to the content of a Grid column.

Embedding controls

Other graphical controls can be shown in Grid cells using the [DISPLAY](#) statement. The value of these controls becomes the [Cell-Data](#) of the cell.

The following program, creates a Grid control that uses Check-Boxes to display and accept data on the first column and Combo-boxes to accept data on the second column:

```

program-id. grid-sample.
working-storage section.
copy "iscrt.def".
copy "isgui.def".
77 MainWindow handle of window.
77 My-Combo-Box handle of combo-box occurs 10.
77 My-Check-Box handle of check-box occurs 10.
77 idx pic 9(2) value 0.
77 keystatus pic 9(5) special-names crt status.
88 esc value 27.
screen section.
01 Screen1.
03 My-Grid grid LINE 2, COL 2, SIZE 70 CELLS, LINES 5,
    DISPLAY-COLUMNS (1, 11, 26, 46), NUM-ROWS 10,
    hscroll vscroll
    event procedure My-Grid-Event-Handler.
procedure division.
Main-Logic.
    display standard graphical window background-low
        handle MainWindow
    display Screen1
    perform varying idx from 1 by 1 until idx > 10
        display Check-Box
            handle My-Check-Box(idx)
            event check-evt
            upon My-Grid(idx, 1)
        display Combo-Box lines 5
            handle My-Combo-Box(idx)
            ITEM-TO-ADD ("Item1", "Item2", "Item3",
                "Item4", "Item5", "Item6")
            notify-selchange
            event combo-evt
            upon My-Grid(idx, 2)
    end-perform
    perform until esc
        accept Screen1 on exception continue end-accept
    end-perform
    perform varying idx from 1 by 1 until idx > 10
        destroy My-Check-Box(idx)
        destroy My-Combo-Box(idx)
    end-perform
stop run
.
```

```

My-Grid-Event-Handler.
    display " grid event-type [" event-type "]"
        upon sysout
.
check-evt.
    display " check-box event-type [" event-type "]"
        upon sysout
.
combo-evt.
    display " combo-box event-type [" event-type "]"
        upon sysout
.

```

In order to create a control on each row of a column, the special value -1 can be used as y coordinate. According to the above example, this syntax

```

perform varying idx from 1 by 1 until idx > 10
    display Check-Box
        handle My-Check-Box(idx)
        event check-evt
        upon My-Grid(idx, 1)
    display Combo-Box lines 5
        handle My-Combo-Box(idx)
        ITEM-TO-ADD ("Item1", "Item2", "Item3",
                    "Item4", "Item5", "Item6")
        notify-selchange
        event combo-evt
        upon My-Grid(idx, 2)
end-perform

```

Could be rewritten as

```

display Check-Box
    handle My-Check-Box(idx)
    event check-evt
    upon My-Grid(-1, 1)
display Combo-Box lines 5
    handle My-Combo-Box(idx)
    ITEM-TO-ADD ("Item1", "Item2", "Item3",
                "Item4", "Item5", "Item6")
    notify-selchange
    event combo-evt
    upon My-Grid(-1, 2)

```

Note: Check-Box, Push-Button and Frame are shown permanently on the grid. Other controls are shown only when the user clicks on the cell. Pressing F4 when the focus is on the Combo-Box within a Grid cell doesn't drop the list of values.

Controls can also be shown only during the editing of a cell by displaying them in the [MSG-BEGIN-ENTRY](#)

event and destroying them in the `MSG-FINISH-ENTRY` event, as shown in the below example:

```
program-id. Grid1.
working-storage section.
copy "iscrt.def".
copy "isgui.def".
77 MainWindow      handle of window.
77 My-Combo-Box    handle of combo-box.
screen section.
01 Screen1.
    03 My-Grid grid LINE 2, COL 2, SIZE 70 CELLS, LINES 5,
        DISPLAY-COLUMNS (1, 11, 26, 46), NUM-ROWS 50,
        event procedure My-Grid-Event-Handler.
procedure division.
Main-Logic.
    display standard graphical window background-low
        handle MainWindow
    display Screen1
    perform until 1 = 2
        accept Screen1 on exception continue end-accept
    end-perform
    stop run
.
My-Grid-Event-Handler.
    evaluate event-type
    when msg-begin-entry
        display Combo-Box lines 5
        handle My-Combo-Box
            ITEM-TO-ADD ("Item1", "Item2", "Item3",
                        "Item4", "Item5", "Item6")
        upon My-Grid(event-data-2, event-data-1)
    when msg-finish-entry
        destroy My-Combo-Box
    end-evaluate
.
```

Note: Controls cannot be displayed over heading cells.

Best practice for Check-Box handling

In order to have a response at every click on the Check-Box shown in Grid cells and avoid the runtime to consider some of the clicks as a go-to-cell actions, the best practice is to assign the Check-Box an Exception-Value and the Self-Act style and intercept the click as an exception of the ACCEPT of the screen. The following sample program demonstrates it:

```
program-id. GridCheck.
working-storage section.
copy "isgui.def".
copy "iscrt.def".
77 crt-status special-names crt status pic 9(5).
77 row pic 999.
01 rec.
   03 c-1 pic 9 value 1.
   03 c-2 pic xxx value "abc".

screen section.
01 screen1.
   03 g grid line 2 col 2 lines 10, size 10 cells
       display-columns (1, 3) virtual-width 9
       data-columns (record-position of c-1,
                     record-position of c-2)
       .

procedure division.
main.
   display standard graphical window.
   display screen1.
   display check-box upon g(-1, 1)
       self-act, exception-value 100.
   modify g record-to-add rec.
   modify g record-to-add rec.
   perform until exit
       accept screen1 exception crt-status continue end-accept
       if crt-status = 100
           inquire g cursor-y row
           display "click on check-box in grid" upon sysout
       end-if
   end-perform.
goback.
```

Properties

The following properties are applicable to the GRID control: Action, Alignment, Background-Color, Bitmap, Bitmap-Number, Bitmap-Trailing, Bitmap-Width, Border-Color, Cell-Background-Color, Cell-Color, Cell-Columns-Span, Cell-Current-Background-Color, Cell-Current-Color, Cell-Current-Font, Cell-Current-Foreground-Color, Cell-Current-Protection, Cell-Data, Cell-Entry-Background-Color, Cell-Entry-Color, Cell-Entry-Foreground-Color, Cell-Font, Cell-Foreground-Color, Cell-Hint, Cell-Protection, Cell-Rows-Span, Cell-Selected-Background-Color, Cell-Selected-Color, Cell-Selected-Foreground-Color, Cells-Selected, Col, Color, Column, Column-Background-Color, Column-Color, Column-Dividers, Column-Font, Column-Foreground-Color, Column-Headings-Height, Column-Headings-Layout, Column-Hiding, Column-Protection, Column-Selected-Background-Color, Column-Selected-Color, Column-Selected-Foreground-Color, Columns-Selected, Css-Base-Style-Name, Css-Style-Name, Cursor-Background-Color, Cursor-Color, Cursor-Foreground-Color, Cursor-Frame-Width, Cursor-X, Cursor-Y, Custom-Data, Data-Columns, Data-Types, Display-Columns, Divider-Color, Drag-Background-Color, Drag-Color, Drag-Foreground-Color, Editor-Show-Always, Enabled, End-Color,

Entry-Reason, Event-List, Exclude-Event-List, Export-File-Format, Export-File-Name, File-Pos, Finish-Reason, Font, Foreground-Color, Heading-Background-Color, Heading-Color, Heading-Divider-Color, Heading-Font, Heading-Foreground-Color, Heading-Menu-Popup, Help-Id, Hidden-Data, Hint, Hscroll-Pos, Id, Insert-Rows, Insertion-Index, Last-Row, Layout-data, Line, Lines, Lm-On-Columns, Mass-Update, Max-Height, Max-Width, Min-Height, Min-Width, Model-To-View-Y, Mouse-Wheel-Scroll, Num-Col-Headings, Num-Row-Headings, Num-Rows, Pop-Up Menu, Pos, Position, Protection, Record-Data, Record-To-Add, Record-To-Delete, Region-Background-Color, Region-Color, Region-Foreground-Color, Reordering-Col-Index, Reset-Grid, Row-Background-Color, Row-Background-Color-Pattern, Row-Capacity, Row-Color, Row-Color-Pattern, Row-Cursor-Background-Color, Row-Cursor-Color, Row-Cursor-Foreground-Color, Row-Dividers, Row-Font, Row-Foreground-Color, Row-Foreground-Color-Pattern, Row-Hiding, Row-Protection, Row-Selected-Background-Color, Row-Selected-Color, Row-Selected-Foreground-Color, Rows-Per-Page, Rows-Selected, Search-Options, Search-Text, Search-Text-In-View, Selection-Mode, Separation, Size, Sort-data, Sort-Types, Start-X, Start-Y, VPadding, View-Cursor-Y, View-To-Model-Y, Virtual-Width, Visible, Vscroll-Pos, X, Y.

Action

A specific action is performed when a value is assigned to this property. The following symbolic values, included in the copy file [isgui.def](#), can be assigned. The table below shows the actions related to each value:

action-copy	The content of the Grid is copied to the clipboard. If Selection-Mode is set to a value greater than 0, only the selected cells are copied. Using Java7 or greater, the text format (font and colors) is copied as well.
action-current-page	The Grid control is emptied, excluding headings, and a MSG-PAGED-NEXTPAGE event is generated. This action only has an effect on Grids with the Paged style.
action-entry	The Grid control enters the Edit Mode and the MSG-BEGIN-ENTRY event is generated.
action-export	The content of the Grid is exported to the file name indicated by Export-File-Name in the format indicated by Export-File-Format .
action-first-page	The program acts as if the user requested the first page of data pushing the appropriate button on the right side of the Grid control or by pressing [Ctrl+Home]. A MSG-PAGED-FIRST event is generated. This action only has an effect on Grids with the Paged style.
action-hide-drag	Removes the color set by the Drag-Color property.
action-last-page	The program acts as if the user requested the last page of data pushing the appropriate button on the right side of the Grid control or by pressing [Ctrl+End]. A MSG-PAGED-LAST event is generated. This action only has an effect on Grids with the Paged style.
action-next	The program acts as if the user requested the next record pushing the appropriate button on the right side of the Grid control or by pressing [DownArrow] when the cursor is on the last record of the Grid control. A MSG-PAGED-NEXT event is generated. This action only has an effect on Grids with the Paged style.
action-next-page	The program acts as if the user requested the next page of data pushing the appropriate button on the right side of the Grid control or by pressing [PageDown]. A MSG-PAGED-NEXTPAGE event is generated. This action only has an effect on Grids with the Paged style.
action-previous	The program acts as if the user requested the previous record pushing the appropriate button on the right side of the Grid control or by pressing [UpArrow] when the cursor is on the first record of the Grid control, excluding headings. A MSG-PAGED-PREV event is generated. This action only has an effect on Grids with the Paged style.

action-previous-page	The program acts as if the user requested the previous page of data pushing the appropriate button on the right side of the Grid control or by pressing [PageUp]. A MSG-PAGED-PREVPAGE event is generated. This action only has an effect on Grids with the Paged style.
action-sort	The Grid control sorts itself following Sort-data settings, and the MSG-BEGIN-SORT event is generated. This action affects the current data in the Grid. If the Grid is empty, nothing happens.

The Action property should never be set in Event procedures.

Example - Modify the action property of a Grid

```
...
procedure division.
...
    modify screen-1-gr-1 action action-entry
...
```

Alignment

This property defines the alignment for each column of the table. Allowed values are:

"L"	The content of the cell is left aligned. Leading spaces are ignored.
"R"	The content of the cell is right aligned. Trailing spaces are ignored.
"C"	The content of the cell is centered. Leading and trailing spaces are ignored.
"U"	The content of the cell is left aligned. Leading spaces are kept.
"H"	The content of the cell rendered as HTML. Note - leading spaces after the <html> tag are trimmed.

Since this setting affects the alignment of every single column, a list of values is needed in order to determine how to align them.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the 1st column is left aligned and the 2nd column is centered. The other columns, if any, will be unaligned, the default.

```
ALIGNMENT = ("L", "C")
```

When set to space or spaces, the list is reset.

When a single value other than space is set, it is appended to the list. This is useful to define a user-defined appearance.

Example - Define column alignments by reading them from a Occurs

```
procedure division
...
modify screen-1-gr-1, alignment = spaces | Resets the list of values
perform varying columnidx from 1 by 1 until columnidx > columncount
    modify screen-1-gr-
1, alignment = columnalignment(columnidx) | Sets the alignment of the next column
end-perform
```

Background-Color

This property allows you to set or retrieve the background color of the Grid control. See "[Color management](#)" for further details.

Example - Define a grid with background and foreground color

```
screen section.
...
03 screen-1-gr-1 Grid
    line 7.6
    column 5.8
    size 28.2 cells
    lines 19.8 cells
    background-color 3
    foreground-color 7
    id 1
    no-box
    column-headings
    row-dividers 1
    heading-font Default-Font
    cursor-frame-width 3
    num-rows 5
    .
```

Bitmap

Assigning a bitmap handle to this property indicates that a bitmap is shown in the cell identified by the *X* and *Y* properties. The value "0" removes the bitmap from the cell.

Example - Modify a grid to include a bitmap on one cell

```
working-storage section.  
77 icon-png pic s9(9) comp-4.  
...  
procedure division.  
...  
    call "w$bitmap" using wbitmap-load "icon.png"  
        giving icon-png.  
...  
    modify screen-1-gr-1 x = 1, y = 2  
        bitmap icon-png  
        bitmap-number 3  
        bitmap-trailing 1  
        bitmap-width 18  
        cell-data "Albert"  
        .  
...
```

Bitmap-Number

This property defines which bitmap (among the ones in the bitmap strip referenced by the [Bitmap](#) property) is to be displayed in the cell identified by the [X](#) and [Y](#) properties.

Example - Modify a grid to include a bitmap on one cell

```
working-storage section.  
77 icon-png pic s9(9) comp-4.  
...  
procedure division.  
...  
    call "w$bitmap" using wbitmap-load "icon.png"  
        giving icon-png.  
...  
    modify screen-1-gr-1 x = 1, y = 2  
        bitmap icon-png  
        bitmap-number 3  
        bitmap-trailing 1  
        bitmap-width 18  
        cell-data "Albert"  
        ....
```

Bitmap-Trailing

This property defines the bitmap position with respect to the text contained in the cell identified by the [X](#) and [Y](#) properties. When set to 0, the default, the bitmap will be on the left side of the text. When set to 1, it will be on the right side.

Example - Modify a grid to include a bitmap on one cell

```
working-storage section.  
77 icon-png pic s9(9) comp-4.  
...  
procedure division.  
...  
    call "w$bitmap" using wbitmap-load "icon.png"  
        giving icon-png.  
...  
    modify screen-1-gr-1 x = 1, y = 2  
        bitmap icon-png  
        bitmap-number 3  
        bitmap-trailing 1  
        bitmap-width 18  
        cell-data "Albert"  
        ....
```

Bitmap-Width

This property defines the width in pixels of the image displayed in the cell identified by the **X** and **Y** properties. The bitmap strip identified by the **Bitmap** property is divided into several smaller images. The width of each image is the value assigned to this property.

Example - Modify a grid to include a bitmap on one cell

```
working-storage section.  
77 icon-png pic s9(9) comp-4.  
...  
procedure division.  
...  
    call "w$bitmap" using wbitmap-load "icon.png"  
        giving icon-png.  
...  
    modify screen-1-gr-1 x = 1, y = 2  
        bitmap icon-png  
        bitmap-number 3  
        bitmap-trailing 1  
        bitmap-width 18  
        cell-data "Albert"  
        ....
```

Border-Color

This property allows you to set or retrieve the border color of the GRID control. See "[Color management](#)" for further details. The border color is applicable only to controls with the **Boxed** style.

Example - define a grid with red border

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   border-color 5  
   .  
...
```

Cell-Background-Color

This property allows you to set or retrieve the background color of the cell identified by the **X** and **Y** properties. See "[Color management](#)" for further details.

Example - Modify a grid to set the background color of a cell

```
procedure division.  
...  
  modify screen-1-gr-1 x = 1, y = 2  
    bitmap icon-png  
    bitmap-number 3  
    bitmap-trailing 1  
    bitmap-width 18  
    cell-background-color 3  
    cell-foreground-color 7  
    cell-data "Albert"  
    .  
...
```

Cell-Color

This property allows you to set or retrieve the color of the cell identified by the **X** and **Y** properties. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details. When this property is set to zero, the cell inherits the row color.

Example - Modify a grid to set the background color of a cell

```
procedure division.  
...  
  modify screen-1-gr-1 x = 1, y = 2  
    bitmap icon-png  
    bitmap-number 3  
    bitmap-trailing 1  
    bitmap-width 18  
    cell-color 232  
    cell-data "Albert"  
    .  
...
```

Cell-Columns-Span

This property allows to group several cells on the x-axes in the column heading.

The value of this property specifies how many cells should be merged together on the x-axes starting from the column identified by the [X](#) property in the row identified by the [Y](#) property. The row must be part of the column heading.

It is a virtual join that the runtime simulates by hiding the vertical border. The number of columns doesn't change for the program.

Example - In a grid where the first three rows are column headings, merge column number 2 and 3 within the second row

```
procedure division.  
...  
    modify screen-1-gr-1(2, 2) cell-columns-span 2  
...  

```

Cell-Current-Background-Color

This property allows you to retrieve the background color of the cell identified by the [X](#) and [Y](#) properties. See "[Color management](#)" for further details.

Example - Get the background color of a cell

```
procedure division.  
...  
    inquire screen-1-gr-1(2, 1)  
        cell-current-background-color ws-cell-b-color  
...  

```

Cell-Current-Color

This property allows you to retrieve the current color of the cell identified by the [X](#) and [Y](#) properties. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Get the color of a cell

```
procedure division.  
...  
    inquire screen-1-gr-1(2, 1)  
        cell-current-color ws-cell-color  
...  

```

Cell-Current-Font

This property allows you to retrieve the font of the cell identified by the [X](#) and [Y](#) properties.

Example - Get the font of a cell

```
procedure division.  
...  
    inquire screen-1-gr-1(2, 1)  
           cell-current-font ws-cell-font  
...  

```

Cell-Current-Foreground-Color

This property allows you to retrieve the foreground color of the cell identified by the [X](#) and [Y](#) properties. See "[Color management](#)" for further details.

Example - Get the foreground color of a cell

```
procedure division.  
...  
    inquire screen-1-gr-1(2, 1)  
           cell-current-foreground-color ws-cell-f-color  
...  

```

Cell-Current-Protection

This property allows you to retrieve the current protection setting of the cell identified by the [X](#) and [Y](#) properties. See [Cell-Protection](#) property for details about protection.

Example - Get the protection status of a grid cell

```
...  
procedure division.  
...  
    inquire screen-1-gr-1(2, 1)  
           cell-current-protection ws-prot  
...  

```

Cell-Data

This property allows you to set or retrieve the text of the cell identified by the [X](#) and [Y](#) properties. HTML is allowed, see the [Alignment](#) property.

Example - Get the data of a grid cell

```
...  
procedure division.  
...  
    inquire screen-1-gr-1(2, 1) cell-data ws-data  
...  

```

Cell-Entry-Background-Color

This property allows you to set or retrieve the background color of the cell that contains the cursor when such cell is in edit mode. See "[Color management](#)" for further details.

Example - Set the background color during entry mode for a grid

```
procedure division.  
...  
    modify screen-1-gr-1 cell-entry-background-color 7  
...
```

Cell-Entry-Color

This property allows you to set or retrieve the color of the cell that contains the cursor when such cell is in edit mode. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Set the colors during entry mode for a grid

```
procedure division.  
...  
    modify screen-1-gr-1 cell-entry-color 480  
..
```

Cell-Entry-Foreground-Color

This property allows you to set or retrieve the foreground color of the cell that contains the cursor when such cell is in edit mode. See "[Color management](#)" for further details.

Example - Set the text color during entry mode for a grid

```
procedure division.  
...  
    modify screen-1-gr-1 cell-entry-foreground-color 3  
..
```

Cell-Font

This property allows you to set or retrieve the font of the cell identified by the [X](#) and [Y](#) properties. When this property is set to zero, the cell inherits the row font.

Example - Set the font of a cell

```
procedure division.  
...  
  initialize wfont-data tahoma-10v0.  
  move 10 to wfont-size.  
  move "Tahoma" to wfont-name.  
  set wfont-bold to false.  
  set wfont-italic to false.  
  set wfont-underline to false.  
  set wfont-strikeout to false.  
  set wfont-fixed-pitch to false.  
  call "w$font" using wfont-get-font tahoma-10v0 wfont-data.  
...  
  modify screen-1-gr-1 x = 1, y = 2  
    bitmap icon-png  
    bitmap-number 3  
    bitmap-trailing 1  
    bitmap-width 18  
    cell-color 232  
    cell-font Tahoma-10v0  
    cell-data "Albert"  
    .  
...
```

Cell-Foreground-Color

This property allows you to set or retrieve the foreground color of the cell identified by the [X](#) and [Y](#) properties. See "[Color management](#)" for further details.

Example - Modify a grid to set the background color of a cell

```
procedure division.  
...  
  modify screen-1-gr-1 x = 1, y = 2  
    bitmap icon-png  
    bitmap-number 3  
    bitmap-trailing 1  
    bitmap-width 18  
    cell-background-color 3  
    cell-foreground-color 7  
    cell-data "Albert"  
    .  
...
```

Cell-Hint

This property allows you to set or retrieve the hint text of the cell identified by the [X](#) and [Y](#) properties. Grid headings can host at best, one hint per column. Hints on single cells are not supported by the AWT interface.

Example - Set the hint text of a cell

```
procedure division.  
...  
    modify screen-1-gr-1(2, 1) cell-hint "Cell containing name"  
...
```

Cell-Protection

This property allows you to set or retrieve the protection setting of the cell identified by the **X** and **Y** properties.

Valid values are:

0	The cell is not protected at all. The user can access the cell and modify its value. This is the default setting.
1	The cell is read-only. The user can move into the cell, but cannot change its content.
2	The cell is protected. Any access to the cell is denied. If the user tries to move into a protected cell with the keyboard, it is skipped and the cursor goes to the next valid cell. If protected cell is clicked, no action occurs.

Example - Make a cell read only

```
procedure division.  
...  
    modify screen-1-gr-1(2, 1) cell-protection 1  
...
```

Cell-Rows-Span

This property allows to group several cells on the y-axis in the column heading.

The value of this property specifies how many cells should be merged together on the y-axis starting from the row identified by the **Y** property in the column identified by the **X** property.

Only the first of the merged cells can be used by the program.

Example - In a grid where the first three rows are column headings, merge rows number 2 and 3 within the first column

```
procedure division.  
...  
    modify screen-1-gr-1(2, 1) cell-rows-span 2  
...
```

Cell-Selected-Background-Color

This property allows you to set or retrieve the background color of the selected cells when **Selection-Mode** includes *grsm-cell-selection*. See "[Color management](#)" for further details.

Example - Set the background color for selected cells

```
procedure division.  
...  
    modify screen-1-gr-1 cell-selected-background-color 7  
...
```

Cell-Selected-Color

This property allows you to set or retrieve the color of the selected cells when [Selection-Mode](#) includes *grsm-cell-selection*. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Set the colors for selected cells

```
procedure division.  
...  
    modify screen-1-gr-1 cell-selected-color 480  
..
```

Cell-Selected-Foreground-Color

This property allows you to set or retrieve the foreground color of the selected cells when [Selection-Mode](#) includes *grsm-cell-selection*. See "[Color management](#)" for further details.

Example - Set the foreground color for selected cells

```
procedure division.  
...  
    modify screen-1-gr-1 cell-selected-foreground-color 7  
...
```

Cells-Selected

This property allows to retrieve the list of selected cells when [Selection-Mode](#) includes *grsm-cell-selection*.

The list of cells is returned in the format: row1,col1 row2,col2 ... rowN,colN.

Example - Retrieve the list of selected cells and show it to the user

```
working-storage section.  
77 selected-cells-list pic x any length.  
  
procedure division.  
...  
    inquire screen1-gr-1 cells-selected selected-cells-list.  
    display message selected-cells-list.  
...
```

[Col | Column | Pos | Position]

This property allows you to specify the Grid control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Grid control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Grid control will be relative to the ending position of the prior Screen Section item.

When the Grid control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03  Label, COL 2, SIZE 12, (more screen options).  
03  Grid, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a grid at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
  03 screen-1-gr-1 Grid  
    line 8.0  
    column 5.0  
    color 7  
    size 45.0 cells  
    id 2  
    .
```

Color

This property allows you to set or retrieve the color of the Grid control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a grid with background and foreground color in one property

```
screen section.  
...  
  03 screen-1-gr-1 Grid  
    line 7.6  
    column 5.8  
    size 28.2 cells  
    lines 19.8 cells  
    color 294  
    id 1  
    no-box  
    column-headings  
    row-dividers 1  
    heading-font Default-Font  
    cursor-frame-width 3  
    num-rows 5  
    .
```

Column-Background-Color

This property allows you to set or retrieve the background color of the column identified by the **X** property. See "[Color management](#)" for further details.

Example - Modify a grid, set the background and foreground color of one column

```
procedure division.  
...  
    modify screen-1-gr-1 x = 3  
        column-background-color 13  
        column-foreground-color 0  
        .  
...  

```

Column-Color

This property allows you to set or retrieve the color of the column identified by the **X** property. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details. When this property is set to zero, the column inherits the grid color.

Example - Set the color of a grid column

```
procedure division.  
...  
    modify screen-1-gr-1 x = 2  
        column-color 130  
        .  
...  

```

Column-Dividers

This property defines the thickness of each line between columns. Valid values are:

-1	Resets any value previously specified.
0	No line is drawn between columns.
> 0	A line is drawn and the number represents its thickness, in pixels.

Since this setting affects the appearance of the line between two columns, a list of values is needed in order to determine how to draw the lines between each pair of columns.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the line between the 1st and 2nd column is one pixel wide, the line between 2nd and 3rd column is 2 pixels wide and the line between 3rd and 4th column is 3 pixels wide. The lines between the other columns, if any, will be one pixel wide, the default.

```
COLUMN-DIVIDERS = ( 1, 2, 3 )
```

When a single value greater than or equal to zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Example - Set column dividers by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-gr-1, column-dividers = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-gr-1, column-  
dividers = dividerwidth(columnidx) | sets the width of the next divider  
end-perform
```

Column-Font

This property allows you to set or retrieve the font of the column identified by the [X](#) property. When this property is set to zero, the column inherits the grid font.

Example - Set the font of a grid column

```
procedure division.  
...  
    modify screen-1-gr-1 x = 2  
        column-font Courier-New-10v0  
    .  
...  
.
```

Column-Foreground-Color

This property allows you to set or retrieve the foreground color of the column identified by the [X](#) property. See "[Color management](#)" for further details.

Example - Modify a grid, set the background and foreground color of one column

```
procedure division.  
...  
    modify screen-1-gr-1 x = 3  
        column-background-color 13  
        column-foreground-color 0  
    .  
...  
.
```

Column-Headings-Height

This property sets the height of each row in the column headings.

The value must be greater than 0 and can have decimals (e.g. 0,3).

By default rows in the column headings have an height of 1. It's useful to increase this height when you need to display text on multiple lines in the column headings having "H" as [Alignment](#) or when you have vertical headings specified by [Column-Headings-Layout](#).

This property replaces Row-Headings-Line-Height that is no more supported.

Example - Modify a grid to set the heading line height

```
procedure division.  
...  
    modify screen-1-gr-1 column-headings-height 2  
...
```

Column-Headings-Layout

This property defines the layout of each column by adding together one or more of the following values, defined in [isgui.def](#)

Constant name	Value	Description
grchl-horizontal	0	The text is displayed horizontally, from left to right.
grchl-vertical-left	1	The text is displayed vertically, from bottom to top.
grchl-vertical-right	2	The text is displayed vertically, from top to bottom.
grchl-align-center	4	The text is centered in the cell.
grchl-align-left	8	The text is left aligned in the cell.
grchl-align-right	16	The text is right aligned in the cell.
grchl-align-top	32	The text is placed at the top of the cell.
grchl-align-bottom	64	The text is placed at the bottom of the cell.

Since this setting affects each column, a list of values is needed in order to determine how to draw them.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies a vertical heading on the 1st column and horizontal heading on the 2nd column.

```
COLUMN-HEADINGS-LAYOUT = (1, 0)
```

When a single value greater than or equal to zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Setting the property to -1 resets the list.

Bitmaps shown in the heading cells are not rotated along with the text.

Example - Modify a grid to set vertical headings

```
procedure division.  
...  
    modify screen-1-gr-1  
        column-headings-layout ( 1 1 1 )  
        .  
...
```

Column-Hiding

This property hides the column identified by the **X** property. The column is still available to the program, but is no longer visible to the user. Valid values are:

0	Shows a column.
1	Hides a column.

Example - Hide the third column of a grid

```
procedure division.  
...  
    modify screen-1-gr-1 x=3 column-hiding=1
```

Column-Protection

This property allows you to set or retrieve the protection setting of the column identified by the **X** property.

Valid values are:

0	The column is not protected at all. The user can access the cells and modify their value. This is the default setting.
1	The column is read-only. The user can move into any cell in the column, but cannot change its content.
2	The column is protected. Any access to the cells in that column is denied. If the user tries to move to a protected column with the keyboard, it is skipped and the cursor goes to the next valid column. If protected column is clicked, no action occurs.

Example - Protect a column of a grid

```
procedure division.  
...  
    modify screen-1-gr-1 x = 2 column-protection 1  
...
```

Column-Selected-Background-Color

This property allows you to set or retrieve the background color of the selected columns when **Selection-Mode** includes *grsm-column-selection*. See "[Color management](#)" for further details.

Example - Set the background color for selected columns

```
procedure division.  
...  
    modify screen-1-gr-1 column-selected-background-color 7  
...
```

Column-Selected-Color

This property allows you to set or retrieve the color of the selected columns when [Selection-Mode](#) includes *grsm-column-selection*. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Set the colors for selected columns

```
procedure division.  
...  
    modify screen-1-gr-1 column-selected-color 480  
..
```

Column-Selected-Foreground-Color

This property allows you to set or retrieve the foreground color of the selected columns when [Selection-Mode](#) includes *grsm-column-selection*. See "[Color management](#)" for further details.

Example - Set the foreground color for selected columns

```
procedure division.  
...  
    modify screen-1-gr-1 column-selected-foreground-color 7  
...
```

Columns-Selected

This property allows to set or retrieve the list of selected columns when [Selection-Mode](#) includes *grsm-column-selection*.

The list of columns is specified in the format: column1 column2 ... columnN.

Example - Retrieve the list of selected columns and show it to the user

```
working-storage section.  
77 selected-cols-list pic x any length.  
  
procedure division.  
...  
    inquire screen1-gr-1 columns-selected selected-cols-list.  
    display message selected-cols-list.  
...
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a grid with `css-base-style-name`, applicable with WD2

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   css-base-style-name "css-grid"  
   no-box  
   column-headings  
   row-dividers 1  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a grid with `css-style-name`, applicable with WD2

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   css-style-name "css-grid"  
   no-box  
   column-headings  
   row-dividers 1  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Cursor-Background-Color

This property allows you to set or retrieve the background color of the cell that contains the cursor. See "[Color management](#)" for further details.

Example - Define a grid with background and foreground color for the cursor cell

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-background-color 10  
   cursor-foreground-color 8  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Cursor-Color

This property allows you to set or retrieve the color of the cell that contains the cursor. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a grid with the color of the cursor cell

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-color 393  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Cursor-Foreground-Color

This property allows you to set or retrieve the foreground color of the cell that contains the cursor. See "[Color management](#)" for further details.

Example - Define a grid with background and foreground color for the cursor cell

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-background-color 10  
   cursor-foreground-color 8  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Cursor-Frame-Width

This property allows you to set the way the cell containing the cursor will be highlighted. Valid values are:

<0	A dotted line is drawn inside the cell. The distance between the dotted line and the border is the absolute value of the value specified. The value -2 means that a dotted line will be drawn inside the cell, two pixels from the border.
0	No border is drawn.
> 0	A black border is drawn. The value specified represents its thickness, in pixels.

By default, the cursor frame width is 2.

Example - Define a grid with the width of the cursor frame

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-background-color 10  
   cursor-foreground-color 8  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Cursor-X

This property allows you to set or retrieve which column the cursor is placed in.

Example - Get the X and Y position of the cell where the cursor is

```
procedure division.  
...  
   inquire screen-1-gr-1 cursor-X ws-X cursor-Y ws-Y  
...  

```

Cursor-Y

This property allows you to set or retrieve which row the cursor is placed in.

Example - Get the X and Y position of the cell where the cursor is

```
procedure division.  
...  
   inquire screen-1-gr-1 cursor-X ws-X cursor-Y ws-Y  
...  

```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a grid

```
procedure division.  
...  
    modify screen-1-gr-1 custom-data "Screen 1 custom data"  
...
```

Data-Columns

The data in a row of the Grid control can be set or retrieved using either a single alphanumeric or a group variable. This property describes the starting position of the information in that variable for each column, starting at 1. The ending position cannot be directly set, it is the character before the first character of the next column.

For example, suppose we have a three-column grid, with the following headers: First Name, Last Name, City. The buffer necessary to add data to this grid should be structured in a way similar to this:

```
01  Grid-Data.  
    03  First_Name  pic x(20).  
    03  Last_Name   pic x(30).  
    03  City        pic x(50).
```

The DATA-COLUMNS property should be set as follows:

```
DATA-COLUMNS = (1, 21, 51)
```

Instead of hard-coded values, it is possible to use the RECORD-POSITION syntax:

```
DATA-COLUMNS = (RECORD-POSITION OF First_Name,  
                 RECORD-POSITION OF Last_Name,  
                 RECORD-POSITION OF City)
```

This syntax avoids problems due to the modification of the item size in the buffer.

When using standard alphanumeric items, the offset of data columns is calculated in bytes, not in digits, so you should pay attention if you're using a variable length encoding (e.g. UTF-8) to store data in the grid record buffer.

When using national items, you can't take advantage of the RECORD-POSITION syntax. You need to use values calculated on the items length in digits, For example, for the following group item:

```
01  Grid-Data USAGE-GROUP NATIONAL.  
    03  First_Name  pic N(20).  
    03  Last_Name   pic N(30).  
    03  City        pic N(50).
```

the correct DATA-COLUMNS setting is:

```
DATA-COLUMNS = (1, 21, 51)
```

and not (1, 41, 101) as RECORD-POSITION would return.

Example - Set data columns by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-gr-1, data-columns = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-gr-1, data-columns = w-data-col(columnidx)  
end-perform
```

Data-Types

This property specifies the type and the length of data contained in the Grid control cells. Multiple values can be specified enclosed between parentheses. Values are applied to cells, starting from the first. Data types are represented by single characters optionally followed by a number enclosed between parentheses. This number represents the maximum number of characters the user can enter in the cell. If the character representing the data type is not followed by any number, the user can type a number of characters equal to the column size. The value -1 indicates that the user can type an unlimited number of characters.

Valid values are:

X	Any alphanumeric character.
U	Any alphanumeric character, converted to upper case.
L	Any alphanumeric character, converted to lower case.
9	0 through 9, decimal point, sign, space.
Z	0 through 9, decimal point, sign, space, currency symbols.
I	0 through 9, sign, space. Suited for integer numbers.
P	0 through 9, space. Suited for positive integer numbers.
D	0 through 9, space, slash, hyphen. Suited for dates.
E	0 through 9, space, slash, hyphen and period. Suited for european dates.

Values can be combined to obtain more accurate filtering.

When even more accurate filtering or formatting is needed, you can display an Entry-Field upon the cell.

Data-Types for date and time

Data types "D" and "E" can be followed by the date format string:

```
D, format-string  
E, format-string
```

format-string can contain any of the following characters:

character	meaning	sample values (comma separated)
G	Era designator	AD

character	meaning	sample values (comma separated)
y	Year	2012
M	Month	07, July, Jul
w	Week in year (1-52)	37
W	Week in month (1-4)	3
D	Day in year (1-365)	201
d	Day in month (1-31)	28
F	Day of week in numbers (1-7)	7
E	Day of week in text	Monday, Mon
a	AM/PM marker	AM, PM
H	Hour in day (0-23)	23
k	Hour in day (1-24)	24
K	Hour in AM/PM (0-11)	11
h	Hour in AM/PM (1-12)	12
m	Minute in hour (0-59)	30
s	Second in minute (0-59)	30
S	Millisecond in second (0-999)	567
z	Timezone General	Pacific Standard Time, PST, GMT-08:00
Z	Timezone RFC 822	-0800

Example:

```
"D, yyyy/MM/dd"
```

Example - Modify a grid to set its column types, alphanumeric, alphanumeric and date

```
procedure division.
...
  modify screen-1-gr-1
    column-dividers ( 1 1 1 )
    data-columns ( 1 17 25 )
    display-columns ( 1 21 29 )
    ( 5 5 5 )
    alignment ( "U" "U" "U" )
    data-types ( "X" "X" "D, yyyy/MM/dd" )
    editor-show-always ( 0 0 0 )
    .
...

```

Display-Columns

This property can be used to set or retrieve the starting position of each column in the Grid control. As a consequence, it defines the number of columns. The width of each column depends on the starting position of the next column. The width of the last column can be defined with the [Virtual-Width](#) property. A column cannot be larger than the Grid width, so the Grid [Size](#) specifies also the maximum size of a column.

Since this property must be set for each column, a list of values is needed in order to determine the starting position of each column.

When values are enclosed between parentheses, a new list is defined at once. The snippet below defines a grid with three columns, 10 characters wide. The first column always starts at 1.

```
DISPLAY-COLUMNS = (1, 11, 21)
VIRTUAL-WIDTH    = 30
```

Setting this property to 0 resets the list.

When a single value greater than zero is set, it is appended to the list. This is useful to define a user-defined appearance.

```
MODIFY MY_GRID, DISPLAY-COLUMNS = 0 | Resets the list of values
MOVE 1 TO ColumnPosition
PERFORM VARYING ColumnIdx FROM 1 BY 1 UNTIL ColumnIdx > ColumnCount
    MODIFY MY_GRID, DISPLAY-
COLUMNS = ColumnPosition | Sets the position of the next column. The first time is 1.
    ADD CustomColumnSize(ColumnIdx) TO ColumnPosition
END-PERFORM
SUBTRACT 1 FROM ColumnPosition
MODIFY MY_GRID, VIRTUAL-WIDTH = ColumnPosition
```

When inquired, this property returns a buffer with the complete list of values, separated by a spaces. The value contained in that buffer can be used to restore the columns width with a single statement.

```
MODIFY MY_GRID, DISPLAY-COLUMNS = (1, 11, 21)
INQUIRE MY_GRID, DISPLAY-COLUMNS IN Buffer | Buffer contains "1 11 21"
MOVE "1 21 31" TO Buffer
MODIFY MY_GRID, DISPLAY-COLUMNS = Buffer | The first column is now 20 characters wide.
```

Example - Modify a grid to set column offsets

```
procedure division.
...
    modify screen-1-gr-1
        display-columns ( 1 21 29 )
        .
...

```

Divider-Color

This property allows you to set or retrieve the color of the Grid control dividers. See "[Color management](#)" for further details.

Example - Define a grid with divider color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-color 393  
   divider-color 5  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Drag-Background-Color

This property allows you to set or retrieve the background color of the drag area (see the [Drag-Color](#) property for an explanation). See "[Color management](#)" for further details.

Example - Define a grid with drag background color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-color 393  
   drag-background-color 14  
   divider-color 5  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Drag-Color

This property allows you to set or retrieve the color that will be used to paint the dynamic area defined when the user clicks and drags the mouse. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

The purpose of this property is to provide a way of selecting cells without any programming effort. When the

user releases the mouse button, the [MSG-END-DRAG](#) event is fired.

Example - Define a Grid with Drag Color

```
screen section.  
....  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-color 393  
   drag-color 333  
   divider-color 5  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Drag-Foreground-Color

This property allows you to set or retrieve the foreground color of the drag area (see the [Drag-Color](#) property for an explanation). See "[Color management](#)" for further details.

Example - Define a grid with drag foreground color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   no-box  
   column-headings  
   row-dividers 1  
   cursor-color 393  
   drag-background-color 14  
   drag-foreground-color 1  
   divider-color 5  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .
```

Editor-Show-Always

This property defines the editor control visibility within each column. A value of 0 means that the editor control will be shown only during cell editing, while a value of 1 means that the editor control will be always

visible. The default value is 0 for every column.

This property affects the following editor controls: Combo-Box, Date-Entry, Entry-Field and List-Box.

Check-Box and Push-Button are always visible, instead.

Since this setting affects each column, a list of values is needed.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the editor visibility is 0 for the 1st column and 1 for the 2nd column. The editor visibility for the other columns, if any, will be 0, the default.

```
EDITOR-SHOW-ALWAYS = (0, 1)
```

When a single value greater than or equal to zero is set, it is appended to the list.

Setting the property to -1 resets the list.

Example - Set editors visibility by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-gr-1, data-columns = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-gr-1, editor-show-always = editor-vis-flag(columnidx)  
end-perform
```

Example - Modify a 3 columns grid to always show the editor of the 1st column

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 6.2  
   column 5.1  
   size 37.4 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...  
procedure division.  
...  
  modify screen-1-gr-1  
    column-dividers ( 1 1 1 )  
    data-columns ( 1 9 17 )  
    display-columns ( 1 9 17 )  
    separation ( 5 5 5 )  
    alignment ( "U" "U" "U" )  
    data-types ( "X" "X" "X" )  
    editor-show-always ( 1 0 0 )  
    .  
...  
...
```

Enabled

This property assumes a value of "0" if the Grid control is disabled, "1" if it is enabled.

Example - Enable a grid on procedure division

```
...  
procedure division.  
...  
  modify screen-1-gr-1 enabled 1  
...  
...
```

End-Color

This property allows you to set or retrieve the color used in the area extending beyond the last row or column. See "[Color management](#)" for further details.

Example - Define a grid with a gray color in the extended areas

```
screen section.  
...  
  03 screen-1-gr-1 Grid  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    no-box  
    column-headings  
    row-dividers 1  
    end-color 9  
    heading-font Default-Font  
    cursor-frame-width 3  
    num-rows 5  
    .  
...
```

Entry-Reason

This property returns a character that represents how the user has switched to entry mode. It can only be inquired during the [MSG-BEGIN-ENTRY](#) event.

Returned values, defined in [isgui.def](#), are:

x"0D"	grer-enter	The user has pressed the [Enter] key.
x"00"	grer-dblclick	The user double-clicked the cell.
x"01"	grer-del	The user has pressed the [DEL] key.
x"FF"	grer-entry-by-program	The Action property has been set to Action-Entry
any other character		The user has pressed a key, such as a number or a letter. Inquiring this property the character pressed by the user is returned.

Example - Inquire the entry-reason from a Grid

```
procedure division.  
...  
  screen-1-gr-1-evt-msg-begin-entry.  
    inquire screen-1-gr-1 entry-reason ws-er  
    if ws-er = grer-dblclick  
      display message "Entry reason: Double click"  
    end-if  
    .  
...
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-](#)

Event-List property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a grid with a list of events to be excluded

```
...
03 screen-1-gr-1 Grid
   line 5.7
   column 3.1
   size 34.6 cells
   lines 15.5 cells
   id 1
   event-list ( cmd-goto cmd-help )
   exclude-event-list 1
   event procedure screen-1-gr-1-evt-proc
   no-box
   column-headings
   row-dividers 1
   end-color 9
   heading-font Default-Font
   cursor-frame-width 3
   num-rows 5
   .
```

Exclude-Event-List

If this property is set to "1", then none of the events in the **Event-List** property are fired. If this property is set to "0", then only the events listed in the **Event-List** property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a grid with a list of events to be excluded

```
...
03 screen-1-gr-1 Grid
   line 5.7
   column 3.1
   size 34.6 cells
   lines 15.5 cells
   id 1
   event-list ( cmd-goto cmd-help )
   exclude-event-list 1
   event procedure screen-1-gr-1-evt-proc
   no-box
   column-headings
   row-dividers 1
   end-color 9
   heading-font Default-Font
   cursor-frame-width 3
   num-rows 5
   .
```

Export-File-Format

This property specifies the default file format in which Grid data must be exported when the 'Export' item is selected from the context menu (see [Heading-Menu-Popup](#)) or when the [Action](#) property is set to *action-export*.

Possible values (case insensitive) are:

Value	Meaning
XLSX	Excel Workbook
XLS	Excel 97- Excel 2003 Workbook

If the property is not set, then "XLS" is assumed.

The cell format in the generated spreadsheet is set according to the following rules:

- If the cell value contains the decimal and the group separator, the format is set to '#,##0.' plus a '0' for each digit after the decimal separator.
- If the cell value contains the decimal separator but not the group separator, the format is set to '0.' plus a '0' for each digit after the decimal separator.
- If the cell value contains the group separator but not the decimal separator, the format is set to '#,###'.
- If the cell value contains neither decimal nor group separator, no cell format is set.

Example - Define a grid with a XLSX file associated for the export feature

```
...
03 screen-1-gr-1 Grid
   line 5.7
   column 3.1
   size 34.6 cells
   lines 15.5 cells
   id 1
   export-file-name "output.xlsx"
   export-file-format "xlsx"
   event procedure screen-1-gr-1-evt-proc
   no-box
   column-headings
   row-dividers 1
   end-color 9
   heading-font Default-Font
   cursor-frame-width 3
   num-rows 5
   .
```

Export-File-Name

This property specifies the name of the file to which Grid data must be exported when the 'Export' item is selected from the context menu (see [Heading-Menu-Popup](#)) or when the [Action](#) property is set to *action-export*.

If the property is not set, then "isCobolGrid.xls" is assumed.

In thin client environment, the file name is resolved on the client machine.

In WD2 environment, the file name is resolved on the web server machine. If the export was triggered from the context menu, then a Save As dialog is shown to allow the user to download the file.

Example - Define a grid with a XLSX file associated for the export feature

```
...
03 screen-1-gr-1 Grid
   line 5.7
   column 3.1
   size 34.6 cells
   lines 15.5 cells
   id 1
   export-file-name "output.xlsx"
   export-file-format "xlsx"
   event procedure screen-1-gr-1-evt-proc
   no-box
   column-headings
   row-dividers 1
   end-color 9
   heading-font Default-Font
   cursor-frame-width 3
   num-rows 5
   .
```

File-Pos

The content of this numeric property is the grid's record number that matches the current file position in the

corresponding data file. The [Paged](#) style must be set.

The File-Pos value will usually be either the last visible record in the grid or the first non-heading record visible.

For example, suppose that you have a grid with five lines and no headings. When you are moving forward through the file, File-Pos will usually be '5', matching the last record added to the grid.

If you click the Next Record button, the MSG-PAGED-NEXT event will indicate that only one READ NEXT is needed to retrieve the appropriate record. Instead, if you click the Previous Record button, the MSG-PAGED-PREV event will indicate that five READ PREVIOUS statements are needed to get the desired record. In this case, File-Pos will change to '1', indicating that only one READ PREVIOUS is needed to get another previous record while five READ NEXT statements are needed to get the next record.

File-Pos has three special values defined as constants in [isgui.def](#):

paged-at-start	The grid will not generate MSG-PAGED-PREV and MSG-PAGED-PREVPAGE events.
paged-at-end	The grid will not generate MSG-PAGED-NEXT and MSG-PAGED-NEXTPAGE events.
paged-empty	MSG-PAGED-NEXT, MSG-PAGED-NEXTPAGE, MSG-PAGE-PREV, and MSG-PAGED-PREVPAGE are not generated. Since it is possible that more records are added to the file and they could be seen by re-reading the file, this value will still generate MSG-PAGED-FIRST and MSG-PAGED-LAST events.

The grid automatically manages File-Pos using the following rules:

- When a record is added to the grid in the topmost non-heading position, File-Pos is set to that position.
- When a record is added to the grid or past the last grid record, File-Pos is set to that position.
- If you set EVENT-ACTION-FAIL in response to a MSG-PAGED-NEXT event, File-Pos is set to Paged-At-End.
- If you set EVENT-ACTION-FAIL in response to a MSG-PAGED-PREV event, File-Pos is set to Paged-At-Start.
- If you set EVENT-ACTION-FAIL in response to a MSG-PAGED-FIRST or MSG-PAGED-LAST event, File-Pos is set to Paged-Empty.
- If a MSG-PAGED-FIRST event sets EVENT-ACTION (this is the default), File-Pos is set to Paged-At-Start.
- If a MSG-PAGED-LAST event sets EVENT-ACTION, File-Pos is set to Paged-At-End.
- If you reset the grid, File-Pos is set to Paged-Empty. Adding records to the grid will change this value.

The handling described above will correctly handle grids whose data is coming from an indexed data file if you move the file's record pointer only in response to grid events. In cases in which you move the file's record pointer independent of a grid request, you will need to do one of the following:

- Modify File-Pos to reflect the actual record position. You may use File-Pos numbers outside of the range of available grid records if needed. Set File-Pos to '1' to point to the first record in the grid, '0' to point to the record before that, '-1' to point to two records before it, and so on. You can also use numbers larger than the last grid record to indicate a position beyond the end of the grid.
- Reposition the current file pointer to match the File-Pos value. You can do this by reading the appropriate record from the data file again. Note that a START may not be good enough. START positions the file pointer so that the next READ NEXT or READ PREVIOUS returns the selected record; it may not return the record positioned at either side of that record.

- Ignore the positioning information passed into the MSG-PAGED-NEXT and MSG-PAGED-PREV events, and the positioning information supplied by the grid control. Supply your own positioning logic. In this case, File-Pos may be incorrect, but File-Pos is irrelevant at this point because you are not using it.

Example - Enquire the file-pos property from a paged grid

```
procedure division.
...
    inquire screen-1-gr-2 file-pos ws-fp
    display message ws-fp
...
```

Finish-Reason

This property returns a number that represents how the user has left the entry mode. It can only be inquired during the [MSG-FINISH-ENTRY](#) and [MSG-CANCEL-ENTRY](#) events.

Returned values, defined in [isgui.def](#), are:

Standard Value	Extended Value	Constant in isgui.def	Description
-8	-8	grfr-autoterminate	The editing has been automatically terminated
-7	-7	grfr-tab-key	The user has pressed the [Tab] key.
n/a	-11	grfr-shift-tab-key	The user has pressed [SHIFT]+[TAB]
-6	-6	grfr-enter-key	The user has pressed the [Enter] key.
-5	-5	grfr-escape-key	The user has pressed the [Esc] key.
-4	n/a	grfr-navigation-key	The user has pressed a navigation key, such as [Up], [Down], [PageUp] or [PageDown].
n/a	-5	grfr-arrow-right-key	The user has pressed the right arrow key
n/a	-9	grfr-page-up-key	The user has pressed the Page Up key.
n/a	-10	grfr-page-down-key	The user has pressed the Page Down key
n/a	-12	grfr-arrow-up-key	The user has pressed the up arrow key
n/a	-13	grfr-arrow-down-key	The user has pressed the down arrow key
n/a	-14	grfr-arrow-left-key	The user has pressed the left arrow key.
-3	-3	grfr-cell-clicked	The user has clicked a cell of the same grid.
-2	-2	grfr-terminating	The user has clicked another control or another window.
-1	-1	grfr-blank-past-end	The user has entered spaces in the last empty row automatically added by the grid. This can only happen when the Num-Rows property is set to -1.

Standard Value	Extended Value	Constant in isgui.def	Description
<i>any other value</i>			A Termination or an Exception condition has occurred. The property value is set to the Exception or Termination value.

In order to activate the extended values, the configuration property `iscobol.gui.grid.extended_finish_reason (boolean) *` must be set to true.

Example - Inquire the finish reason from a grid

```
...
procedure division.
...
    screen-1-gr-1-evt-msg-finish-entry.
        inquire screen-1-gr-1 finish-reason ws-fr
        if ws-fr = grfr-enter-key
            display message "Finish reason: ENTER"
        end-if
    .
...

```

Font

This property specifies the font used to display the content of the Grid control. It may be used to compute the height and the width of the Grid control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a grid with specific Font

```
working-storage section.  
77 Calibri-10v0 handle of font.  
...  
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   font Calibri-10v0  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...  
*> Prior to display the screen with the grid, load the font in  
*> procedure division using w$font  
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Grid control. See "[Color management](#)" for further details.

Example - Define a grid with foreground and background color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   background-color 3  
   foreground-color 7  
   id 1  
   no-box  
   column-headings  
   row-dividers 1  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Heading-Background-Color

This property allows you to set or retrieve the background color of both column and row headings. See "[Color management](#)" for further details.

Example - Define a grid with heading background and foreground color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-background-color 3  
   heading-foreground-color 7  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Heading-Color

This property allows you to set or retrieve the color of both column and row headings. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a grid with heading color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Heading-Divider-Color

This property defines the color of the Grid control dividers. See ["Color management"](#) for further details.

Example - Define a grid with heading divider color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Heading-Font

This property allows you to set or retrieve the font used for both row and column headings.

Example - Define a grid with specific heading font

```
working-storage section.  
77 Calibri-10v0 handle of font.  
...  
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-rows 5  
   .  
...  
*> Prior to display the screen with the grid, load the font in  
*> procedure division using w$font  
...
```

Heading-Foreground-Color

This property allows you to set or retrieve the foreground color of both column and row headings. See "[Color management](#)" for further details.

Example - Define a grid with heading background and foreground color

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-background-color 3  
   heading-foreground-color 7  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Heading-Menu-Popup

This property allows to set or retrieve the status of the context menu shown on grid columns headings. The context menu allows you to choose between hiding or showing a column, exporting data to xls/xlsx spreadsheets, copying data to the clipboard and searching text in the grid.

The possible values for this property, defined in [isgui.def](#), are:

grhm-no-menu (value 0)	The context menu is not available.
grhm-columns-on-right-click (value 1)	The list of columns is shown by right clicking on the grid heading.
grhm-columns-on-button (value 2)	The list of columns is shown by clicking on the button shown on the top left corner of the grid.
grhm-export-on-right-click (value 4)	The 'Export' option is shown by right clicking on the grid heading.
grhm-export-on-button (value 8)	The 'Export' option is shown by clicking on the button shown on the top left corner of the grid.
grhm-copy-on-right-click (value 16)	The 'Copy to Clipboard' option is shown by right clicking on the grid heading.
grhm-copy-on-button (value 32)	The 'Copy to Clipboard' option is shown by clicking on the button shown on the top left corner of the grid.
grhm-find-on-right-click (value 64)	The 'Find' option is shown by right clicking on the grid heading.
grhm-find-on-button (value 128)	The 'Find' option is shown by clicking on the button shown on the top left corner of the grid.

The above values can be combined in order to obtain the desired effect. Setting the property to the value 63 makes all the items available in the menu shown by right clicking as well as in the menu shown by clicking on the top left corner of the grid.

When the 'Export' option is selected, a Save As dialog is shown to allow the user to choose where to save the exported data. The fields of this dialog are preset according to the properties [Export-File-Name](#) and [Export-File-Format](#). Font and colors are replicated in the exported file, while bitmaps and embedded controls are not.

When the 'Copy to Clipboard' option is selected, the Grid content is copied to the clipboard. If you're using Java 7 or greater, the text style (font and colors) is copied as well. If [Selection-Mode](#) is set to a value greater than 0, then only selected cells are copied to the clipboard.

Example - Define a Grid where the list of columns is shown in the menu that appears by right clicking on the heading, while the export option appears by clicking on the top left corner of the grid, and where the 'Copy to clipboard' option is not available.

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    line 15.5  
    column 2.6  
    size 28.4 cells  
    lines 18.6 cells  
    id 3  
    no-box  
    heading-menu-popup 9  
    column-headings  
    row-dividers 1  
    heading-color 133  
    heading-font Default-Font  
    cursor-frame-width 3  
    num-rows 5  
    .  
...  

```

Help-Id

This property allows you to assign a unique ID to the Grid control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a grid with help id

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   font Calibri-10v0  
   help-id 5040  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Hidden-Data

This property allows you to set or retrieve the hidden value of the cell identified by the **X** and **Y** properties. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to a row or a cell.

Example - Modify a grid cell to include hidden data in it

```
...  
procedure division.  
...  
  modify screen-1-gr-1 x = 1, y = 2  
    hidden-data "Hidden data in Cell (2,1)"  
...  

```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Grid control.

Example - Define a grid with hint text

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   font Calibri-10v0  
   help-id 5040  
   id 1  
   hint "Grid Hint Information"  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   num-rows 5  
   .  
...
```

Hscroll-Pos

This numeric property contains the number of the leftmost column currently visible.

Example - Get the number of the leftmost column currently visible on a grid

```
procedure division.  
...  
   inquire screen-1-gr-1 hscroll-pos ws-hsp  
...  

```

Id

This property allows you to assign a unique ID to the Grid control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a grid with the ID property

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   row-dividers 1  
   end-color 9  
   num-rows 5  
   .  
...
```

Insert-Rows

This numeric property establishes how many blank rows are to be inserted at the position set in the [Insertion-Index](#) property.

Example - Modify a grid to insert 3 rows on position 5

```
...  
procedure division.  
...  
   modify screen-1-gr-1 insertion-index 5 insert-rows 3  
...  

```

Insertion-Index

This numeric property affects the position where a new row is added to a Grid control when the [Record-To-Add](#) property is set. If this property is set to a positive value, the row is inserted immediately before the corresponding record. If "0" is assigned to this property, the row is inserted as last record of the Grid control.

Example - Modify a grid to insert 3 rows on position 5

```
...  
procedure division.  
...  
   modify screen-1-gr-1 insertion-index 5 insert-rows 3  
...  

```

Last-Row

This numeric property contains the number of the last non-empty row in the Grid control.

Example - Inquire a grid to get the number of rows it has

```
...  
procedure division.  
...  
    inquire screen-1-gr-1 last-row ws-lr  
...
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a grid that allows resize in X and Y when the layout manager requires so

```
screen section.  
...  
    03 screen-1-gr-1 Grid  
        line 5.7  
        column 3.1  
        size 34.6 cells  
        lines 15.5 cells  
        id 1  
        layout-data 17  
        event procedure screen-1-gr-1-evt-proc  
        no-box  
        column-headings  
        hscroll  
        row-dividers 1  
        end-color 9  
        heading-color 200  
        heading-divider-color 12  
        cursor-frame-width 3  
        num-rows 5  
        .  
...
```

Line

This property allows you to specify the Grid control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Grid control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Grid control will be relative to the starting position of the prior Screen Section item.

When the Grid control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Grid, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a grid at line 8.0 on the screen section definition

```
...
03 screen-1-gr-1 Grid
   line 8.0
   column 5.0
   color 7
   size 45.0 cells
   lines 25.0 cells
   id 2
.
```

Lines

This property allows you to specify the height of the Grid control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Grid control is still computed in CELLS, but the cell size is based on the font set for the Grid control with the [Font](#) property. If no font has been defined for the Grid control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a grid with height in lines

```
screen section.
...
03 screen-1-gr-1 Grid
   line 5.7
   column 3.1
   size 34.6 cells
   lines 15.5 cells
   id 1
   layout-data 17
   event procedure screen-1-gr-1-evt-proc
   no-box
   column-headings
   hscroll
   row-dividers 1
   end-color 9
   heading-color 200
   heading-divider-color 12
   cursor-frame-width 3
   num-rows 5
.
...
```

Lm-On-Columns

This property specifies the behavior of columns when the window is resized and a layout manager is involved.

(NONE)	the behavior is controlled by the <code>iscobol.gui.grid.lm_on_columns (boolean) *</code> property
0	columns are not resized
1	columns are resized

This property is evaluated only if the Grid has the [Adjustable-Columns](#) style.

Example - Define a grid that allows resizing in columns

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    help-id 5040  
    id 1  
    layout-data 17  
    event procedure screen-1-gr-1-evt-proc  
    no-box  
    column-headings  
    hscroll  
    adjustable-columns  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    heading-color 200  
    heading-divider-color 12  
    cursor-frame-width 3  
    num-rows 5  
    .  
...
```

Mass-Update

Setting this property to "1" inhibits isCOBOL framework to repaint the Grid control every time the program modifies it. This practice is recommended to increase performance when a large number of changes are applied to the Grid control. At the end of the process it is necessary to reset the property to its default value of "0" to see the changes.

Example - Add three records in mass update mode

```
modify screen-1-gr-1 mass-update = 1  
modify screen-1-gr-1 record-to-add gr-rec  
modify screen-1-gr-1 record-to-add gr-rec  
modify screen-1-gr-1 record-to-add gr-rec  
modify screen-1-gr-1 mass-update = 0
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a grid with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a grid with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a grid with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a grid with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Model-To-View-Y

When inquired, this property allows you to retrieve the row number on video corresponding to the row number for the program identified by the **Y** property. This kind of information is useful after a sort, when the order of the rows on video doesn't match with the order of the rows for the program.

Example - Get the row number on video of row 5 from a grid

```
procedure division.  
...  
   inquire screen-1-gr-1(5, 1) model-to-view-y ws-mtv-y  
...  

```

Mouse-Wheel-Scroll

This property specifies how many records must scroll in the grid at each mouse wheel movement. It has effect only if the grid has the [Paged](#) style.

Example - Define the number of rows to scroll for the mouse-wheel

```
procedure division.  
...  
    inquire screen-1-gr-1 model-to-view-y ws-mv-y  
...  

```

Num-Col-Headings

With this numeric property it is possible to establish the number of rows used as column headings when the grid has the [Column-Headings](#) style.

Example - Define a grid with number of column headings

```
...  
03 screen-1-gr-1 Grid  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    no-box  
    column-headings  
    hscroll  
    row-headings  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    cursor-frame-width 3  
    num-col-headings 2  
    num-row-headings 2  
    num-rows 5  
    .  
...  

```

Num-Row-Headings

With this numeric property it is possible to establish the number of columns used as row headings when the grid has the [Row-Headings](#) style.

Example - Define a grid with number of row headings

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   row-headings  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   cursor-frame-width 3  
   num-col-headings 2  
   num-row-headings 2  
   num-rows 5  
   .  
...
```

Num-Rows

This numeric property sets the number of rows the Grid control initially contains.

When set to -1, an empty row is always added at the end of the Grid control.

When the user inserts data in that line, a new empty row is appended. This additional row can be used directly to add more records to the Grid control.

When set to 0, the Grid control contains exactly as many rows as inserted.

When set to a value greater than zero, the Grid control contains that number of rows, no matter how many rows the program tries to add.

Example - Define a grid with number of rows

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   row-headings  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Pop-Up Menu

With this property, it is possible to associate a pop-up menu with the Grid control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a grid with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-gr-1 Grid  
   pop-up menu hmenu  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   row-headings  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   cursor-frame-width 3  
   num-rows 5  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Protection

This property allows you to set or retrieve the protection setting of the Grid control.

Valid values are:

0	the Grid control is not protected at all. The user can access the cells and modify their value. This is the default setting.
1	the Grid control is read-only. The user can move to any of the cells, but cannot change their content.
2	the Grid control doesn't allow you to select any cell, however it can still get the focus.

Example - Modify a grid to be read-only in full

```
procedure division.  
...  
   modify screen-1-gr-1 protection 1  
...  

```

Record-Data

This property allows you to set or retrieve the data contained in the row identified by the [Y](#) property. The [Data-Columns](#) property must be properly set.

Example - Get the full record on row 5 of a grid

```
working-storage section.  
01 ws-record.  
    05 col-1    pic x(8) .  
    05 col-2    pic x(3) .  
    05 col-3    pic x(10) .  
    ...  
procedure division.  
    ...  
    modify screen-1-gr-1 y 5  
    inquire screen-1-gr-1 record-data ws-record  
    ...
```

Record-To-Add

This property allows you to add a new row from a buffer. The [Data-Columns](#) property must be properly set. The row is inserted, according to the value set in the [Insertion-Index](#) property.

Example - Add one record to the grid

```
working-storage section.  
01 ws-record.  
    05 col-1    pic x(8) .  
    05 col-2    pic x(3) .  
    05 col-3    pic x(10) .  
    ...  
procedure division.  
    ...  
    move "C0140" to col-1  
    move "XCM"   to col-2  
    move "Adam"  to col-3  
    modify screen-1-gr-1 insertion-index 2 record-to-add ws-record  
    ...
```

Record-To-Delete

When set to a positive value, the corresponding row is removed. Column headings are considered rows as well.

Example - Remove the third row from a grid:

```
modify screen-1-gr-1 record-to-delete 3
```

Region-Background-Color

This property defines the background color used to paint an area of the Grid control. The area is always rectangular. Before setting the background color of that area, its bounds must be defined. The upper left cell is identified by the [Start-X](#) and [Start-Y](#) properties, while the lower right cell is identified by the [X](#) and [Y](#) properties.

Setting those properties after setting the [Region-Background-Color](#) property is not advised as this may lead to unexpected behaviors. A correct code will always look as shown in the snippet below.

Only one area can be changed at a time. Setting the color to an area with different coordinates does not produce a new colored area, it simply changes its location and color.

See "[Color management](#)" for further details.

Example - Change the background color of a region in the grid

```
modify screen-1-gr-1,
  start-x           = UpperLeftColumn,
  start-y           = UpperLeftRow,
  x                 = LowerRightColumn,
  y                 = LowerRightRow,
  region-background-color = MyColor
```

Region-Color

This property defines the color used to paint an area of the Grid control. The area is always rectangular. Before setting the color of that area, its bounds must be defined. The upper left cell is identified by the [Start-X](#) and [Start-Y](#) properties, while the lower right cell is identified by the [X](#) and [Y](#) properties.

Setting those properties after setting the [REGION-COLOR](#) property is not advised as this may lead to unexpected behaviors. A correct code will always look as shown in the snippet below.

Only one area can be changed at a time. Setting the color to an area with different coordinates does not produce a new colored area, it simply changes its location and color.

Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Change the color of a region in the grid

```
modify screen-1-gr-1,
  start-x           = UpperLeftColumn,
  start-y           = UpperLeftRow,
  x                 = LowerRightColumn,
  y                 = LowerRightRow,
  region-color       = MyColor
```

Region-Foreground-Color

This property defines the foreground color used to paint an area of the Grid control. The area is always rectangular. Before setting the foreground color of that area, its bounds must be defined. The upper left cell is identified by the [Start-X](#) and [Start-Y](#) properties, while the lower right cell is identified by the [X](#) and [Y](#) properties.

Setting those properties after setting the [REGION-FOREGROUND-COLOR](#) property is not advised as this may lead to unexpected behaviors. A correct code will always look as shown in the snippet below.

Only one area can be changed at a time. Setting the color to an area with different coordinates does not produce a new colored area, it simply changes its location and color.

See "[Color management](#)" for further details.

Example - Change the foreground color of a region in the grid

```
modify screen-1-gr-1,  
  start-x          = UpperLeftColumn,  
  start-y          = UpperLeftRow,  
  x                = LowerRightColumn,  
  y                = LowerRightRow,  
  region-foreground-color = MyColor
```

Reordering-Col-Index

This property allows you to set or retrieve the order in which the columns are displayed, regardless of the setting of the [Display-Columns](#) property. Thus, the columns can be swapped by the user without affecting the way the source code is written. The value of this property is alphanumeric and consists in a serie of numbers separated by spaces indicating columns offsets.

Example - Get the new column ordering from a grid

```
procedure division.  
...  
  inquire screen-1-gr-1 reordering-col-index ws-reord-index  
...
```

Reset-Grid

When set to 1, the Grid control is completely emptied.

When set to 2, the Grid control is emptied except for the headings. If row headings are present, the Grid cells are emptied but not physically removed from the control.

When set to 3, the Grid control is emptied except for the column headings. Row headings, if present, are removed.

Example - Modify a grid to reset all its rows but the column heading rows

```
procedure division.  
...  
  modify screen-1-gr-1 reset-grid 3  
...
```

Row-Background-Color

This property allows you to set or retrieve the background color of the row identified by the [Y](#) property. See "[Color management](#)" for further details.

Example - Modify row 3 to set its background color

```
procedure division.  
...  
    modify screen-1-gr-1 y = 3 row-background-color 7  
...  

```

Row-Background-Color-Pattern

This property sets the background row color pattern, as explained in the [Row-Color-Pattern](#) property. See "[Color management](#)" for further details.

Example - Modify a grid to set its background color pattern

```
procedure division.  
...  
    modify screen-1-gr-1  
        row-background-color-pattern 7  
        row-background-color-pattern 3  
        row-background-color-pattern 1  
...  

```

Row-Capacity

This property returns the number of records (rows) that can be displayed entirely within the Grid. This number doesn't usually match with the [Lines](#) property if the grid height is measured in cells. This property is particularly useful for paged grids, where there's no vertical scroll-bar so it's important to have all the rows completely visible.

Example - Retrieve the row capacity of a grid and change the num-rows property accordingly

```
working-storage section.  
77 w-rows-displayable pic 9(9).  
...  
procedure division.  
...  
    inquire screen-1-gr-1 row-capacity w-rows-displayable  
    modify screen-1-gr-1 num-rows w-rows-displayable  

```

Row-Color

This property allows you to set or retrieve the color of the row identified by the [Y](#) property. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details. When this property is set to zero, the row inherits the column color.

Example - Modify the color of the third row in the grid

```
procedure division.  
...  
    modify screen-1-gr-1  
        y 3  
        row-color 480  
...  

```

Row-Color-Pattern

This property sets a color pattern to be applied to the Grid control rows. The first color specified here is used for the first pattern row, the second color of the second pattern row and so on. The pattern is then applied to the whole grid.

To obtain a "zebra" effect, add the following definition to the Grid control:

```
ROW-COLOR-PATTERN = (257, 513)
```

Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Modify a grid to set its background color pattern

```
procedure division.  
...  
    modify screen-1-gr-1  
        row-color-pattern 65  
        row-color-pattern 225  
        row-color-pattern 289  
...  

```

Row-Cursor-Background-Color

This property allows you to set or retrieve the background color of the row that contains the cursor. See "[Color management](#)" for further details.

Example - Modify a grid to set the row cursor background color

```
procedure division.  
...  
    modify screen-1-gr-1 row-cursor-background-color 7  
...  

```

Row-Cursor-Color

This property allows you to set or retrieve the color of the row that contains the cursor. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Modify a grid to set the row cursor foreground and background color

```
procedure division.  
...  
    modify screen-1-gr-1 row-cursor-color 225  
...
```

Row-Cursor-Foreground-Color

This property allows you to set or retrieve the foreground color of the row that contains the cursor. See "[Color management](#)" for further details.

Example - Modify a grid to set the row cursor foreground color

```
procedure division.  
...  
    modify screen-1-gr-1 row-cursor-foreground-color 7  
...
```

Row-Dividers

This property defines the thickness of lines between rows. Valid values are:

-1	Resets any value previously specified.
0	No line is drawn between lines.
> 0	A line is drawn and the number represents its thickness, in pixels.

Since this setting affects the appearance of the line between two rows, a list of values is needed in order to determine how to draw the lines between each pair of rows.

Each time you set this property to a value greater than zero, you set the width of the divider for one row of a record. The first setting applies to the first row, the second to the second row, and so on. The pattern established for one record repeats throughout the grid.

Assuming three rows per record, the following code would create a pattern in which records are divided from each other by a two-pixel border, and the three rows within the record are divided by a one-pixel border:

```
ROW-DIVIDERS (1, 1, 2)
```

Example - Display a grid without separation lines between rows

```
screen section.  
...  
03 screen-1-gr-1 Grid  
  pop-up menu hmenu  
  line 5.7  
  column 3.1  
  size 34.6 cells  
  lines 15.5 cells  
  id 1  
  event procedure screen-1-gr-1-evt-proc  
  no-box  
  column-headings  
  hscroll  
  row-headings  
  lm-on-columns 1  
  row-dividers 0  
  end-color 9  
  cursor-frame-width 3  
  num-rows 5  
  .
```

Row-Font

This property allows you to set or retrieve the font of the row identified by the **Y** property. When this property is set to zero, the row inherits the column font.

Example - Modify one row to set its font

```
working-storage section.  
77 Calibri-10v0 handle of font.  
...  
procedure division.  
...  
*> Prior to modify the grid, load the font with w$font  
...  
  modify screen-1-gr-1 y = 5 row-font Calibri-10v0  
...  

```

Row-Foreground-Color

This property allows you to set or retrieve the foreground color of the row identified by the **Y** property. See "Color management" for further details.

Example - Modify a grid to set the foreground color of a row

```
procedure division.  
...  
  modify screen-1-gr-1 y = 5 row-foreground-color 7
```

Row-Foreground-Color-Pattern

This property sets the foreground row color pattern, as explained in the [Row-Color-Pattern](#) property. See "[Color management](#)" for further details.

Example - Modify a grid to set the foreground color pattern

```
procedure division.  
...  
    modify screen-1-gr-1  
        row-foreground-color-pattern 7  
        row-foreground-color-pattern 8  
        row-foreground-color-pattern 9  
...
```

Row-Hiding

This property hides the row identified by the [Y](#) property. The row is still available to the program, but is no longer visible to the user. Valid values are:

0	Shows a row.
1	Hides a row.

This property and the style [Adjustable-Rows](#) cannot coexist.

Example - Hide the third row in a grid

```
modify screen-1-gr-1 y=3 row-hiding=1
```

Row-Protection

This property allows you to set or retrieve the protection setting of the row identified by the [Y](#) property.

Valid values are:

0	The row is not protected at all. The user can access the cells and modify their value. This is the default setting.
1	The row is read-only. The user can move into any of the cells in the row, but cannot change their contents.
2	The row is protected. Any access to the cells in that row is denied. If the user tries to move into a protected row with the keyboard, it is skipped and the cursor goes to the next valid row. If a protected row is clicked, no action occurs.

Example - Modify a grid to protect one of its rows

```
procedure division.  
...  
    modify screen-1-gr-1 row-protection 1  
...
```

Row-Selected-Background-Color

This property allows you to set or retrieve the background color of the selected rows when [Selection-Mode](#) includes *grsm-row-selection*. See "[Color management](#)" for further details.

Example - Set the background color for selected rows

```
procedure division.  
...  
    modify screen-1-gr-1 row-selected-background-color 7  
...
```

Row-Selected-Color

This property allows you to set or retrieve the color of the selected rows when [Selection-Mode](#) includes *grsm-row-selection*. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Set the colors for selected rows

```
procedure division.  
...  
    modify screen-1-gr-1 row-selected-color 480  
..
```

Row-Selected-Foreground-Color

This property allows you to set or retrieve the foreground color of the selected rows when [Selection-Mode](#) includes *grsm-row-selection*. See "[Color management](#)" for further details.

Example - Set the foreground color for selected rows

```
procedure division.  
...  
    modify screen-1-gr-1 row-selected-foreground-color 7  
...
```

Rows-Per-Page

This property has effect only in Web Direct 2.0 (WD2) environment. When set to a value greater than zero, the Grid displays n records at a time (where n is the value of the property) and a navigation bar is shown at the bottom of the Grid allowing the user to navigate to another page of records. The heading rows are not counted in the number of rows per page.

Example - Create a Grid that shows 10 records at a time

```
screen section.  
...  
03 screen-1-gr-1 Grid  
  pop-up menu hmenu  
  line 5.7  
  column 3.1  
  size 34.6 cells  
  lines 15.5 cells  
  id 1  
  event procedure screen-1-gr-1-evt-proc  
  no-box  
  column-headings  
  hscroll  
  row-headings  
  lm-on-columns 1  
  row-dividers 0  
  end-color 9  
  cursor-frame-width 3  
  rows-per-page 10  
  .
```

Rows-Selected

This property allows to set or retrieve the list of selected rows when [Selection-Mode](#) includes *grsm-row-selection*.

The list of rows is specified in the format: row1 row2 ... rowN.

Example - Retrieve the list of selected rows and show it to the user

```
working-storage section.  
77 selected-rows-list pic x any length.  
  
procedure division.  
...  
  inquire screen1-gr-1 rows-selected selected-rows-list.  
  display message selected-rows-list.  
...
```

Search-Options

This property affects the way the information stored in the Grid control is searched. No search is done until the [Search-Text](#) property is set.

The argument passed to this property is the group item GRID-SEARCH-OPTIONS defined in [isgui.def](#). Before using that item, it must be initialized with the INITIALIZE Statement. The following variables or conditions can be set to affect the search behavior:

GRID-SEARCH-FORWARDS

When set to true, the default, the information is searched forwards, otherwise the information is searched backwards.

GRID-SEARCH-WRAP

When set to true, the default, the search does not terminate when the last cell (or the first cell in the case GRID-SEARCH-FORWARDS is set to false) is reached, but automatically continues from the first (or last) cell until the starting cell is reached.

GRID-SEARCH-IGNORE-CASE

When set to true, the default, a case-insensitive search is performed.

[*GRID-SEARCH-MATCH-ANY* | *GRID-SEARCH-MATCH-LEADING* | *GRID-SEARCH-MATCH-ALL*]

GRID-SEARCH-MATCH-ANY	A substring search is performed. This is the default.
GRID-SEARCH-MATCH-LEADING	The search succeeds when the data begins with the searched text.
GRID-SEARCH-MATCH-ALL	The search succeeds only when the data matches the searched text exactly.

[*GRID-SEARCH-VISIBLE* | *GRID-SEARCH-HIDDEN* | *GRID-SEARCH-ALL-DATA*]

GRID-SEARCH-VISIBLE	The search affects only the visible data put in the cells by the user or by the program with the Cell-Data , Insert-Rows , Record-Data or Record-To-Add properties. This is the default.
GRID-SEARCH-HIDDEN	The search affects only the hidden data put in the cells by the program with the Hidden-Data property.
GRID-SEARCH-ALL-DATA	The search affects both visible and hidden data.

GRID-SEARCH-SKIP-CURRENT

When set to true, the current cell is skipped and the search starts from the next one. The default value is true.

GRID-SEARCH-MOVES-CURSOR

When set to true, the cursor is automatically moved to the cell containing the searched text. The default value is false.

GRID-SEARCH-COLUMN

When set to a value greater than zero, the search is performed only in that column. The default value zero means that all columns are to be searched.

Example - Modify a grid to set the search options prior to search

```
working-storage section.  
copy "isgui.def".  
01 search-result pic 9.  
...  
procedure division.  
    set grid-search-forwards      to true  
    set grid-search-wrap          to true  
    set grid-search-ignore-case   to true  
    set grid-search-match-any     to true  
    set grid-search-skip-current  to false  
    set grid-search-moves-cursor  to true  
    set grid-search-all-columns  to true  
    modify h-grid, search-options grid-search-options.  
    modify h-grid (1,1)  
        search-text "part"  
        giving search-result  
        .  
    ...
```

Search-Text

As soon as this property is set to a value, the Grid control is searched for that value, according to the options set in the [Search-Options](#) property.

The search begins at the cell identified by the [X](#) and [Y](#) properties. Cells that are hidden due to [Row-Hiding](#) and [Column-Hiding](#) are not considered by the search.

If the search succeeds, the [X](#) and [Y](#) properties are updated to match the coordinates of the cell that contains the searched text.

After the search, one of the following values is returned:

0	Search failed. The X and Y properties are left unchanged.
1	Search succeeded.
2	Search succeeded, but it reached the end (or the top) of the Grid control and continued from the top (or the end).

Example - Search for the string "Phoenix" and advise the user if the string is not found

```
modify screen-2-gr-1 search-text "Phoenix" giving search-result.  
if search-result = 0  
    display message "Not found" icon mb-error-icon  
end-if.
```

Search-Text-In-View

As soon as this property is set to a value, the Grid control is searched for that value, according to the options set in the [Search-Options](#) property.

The search begins at the cell identified by the [X](#) and [Y](#) properties. Cells that are hidden due to [Row-Hiding](#) and

Column-Hiding are not considered by the search.

If the search succeeds, the **X** and **Y** properties are updated to match the coordinates of the cell that contains the searched text.

Unlike **Search-Text**, Search-Text-In-View looks for the text in the Grid view instead of looking in the model so, if the user reordered or sorted the columns, the search will be performed on the new Grid layout.

After the search, one of the following values is returned:

0	Search failed. The X and Y properties are left unchanged.
1	Search succeeded.
2	Search succeeded, but it reached the end (or the top) of the Grid control and continued from the top (or the end).

Example - Search for the string "Phoenix" and advise the user if the string is not found

```
modify screen-2-gr-1 search-text-in-view "Phoenix" giving search-result.  
if search-result = 0  
    display message "Not found" icon mb-error-icon  
end-if.
```

Selection-Mode

This property activates the ability to select multiple rows, columns and cells in the Grid.

The possible values, defined in **isgui.def**, are:

grsm-no-selection (value 0)	It's not possible to select multiple rows or columns.
grsm-single-selection (value 1)	It's possible to select a single item (row, column or cell) at a time.
grsm-single-interval-selection (value 2)	It's possible to select more contiguous items (rows, columns or cells) at a time.
grsm-multiple-interval-selection (value 4)	It's possible to select more items (rows, columns or cells) at a time, even if they're not contiguous.
grsm-row-selection (value 8)	When selecting a cell, the whole row is selected.
grsm-column-selection (value 16)	When selecting a cell, the whole column is selected.
grsm-cell-selection (value 32)	When selecting a cell, that cell is selected.

The above values can be combined in order to obtain the desired effect. For example, if you wish to allow the user to select multiple contiguous rows, add grsm-row-selection (value 8) to grsm-single-interval-selection (value 2), and you will obtain that the value for the Selection-Mode property is 10.

The selection of multiple items can be performed either by dragging the mouse pointer over the Grid or by holding CTRL (for multiple interval selection) or SHIFT (for single interval selection) on the keyboard and then use arrow keys or click with the mouse.

The color of the selected items is specified by the properties [Cell-Selected-Background-Color](#), [Cell-Selected-Color](#), [Cell-Selected-Foreground-Color](#), [Column-Selected-Background-Color](#), [Column-Selected-Color](#), [Column-Selected-Foreground-Color](#), [Row-Selected-Background-Color](#), [Row-Selected-Color](#) and [Row-Selected-Foreground-Color](#).

The list of the selected items can be retrieved by inquiring the properties [Cells-Selected](#), [Columns-Selected](#) and [Rows-Selected](#).

Example - Define a grid where multiple rows can be selected:

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   pop-up menu hmenu  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   no-box  
   column-headings  
   hscroll  
   row-headings  
   lm-on-columns 1  
   row-dividers 0  
   end-color 9  
   cursor-frame-width 3  
   num-rows 5  
   selection-mode 12  
   row-selected-color 480  
   .
```

Separation

This property defines the amount of blank space at the end of each column, in tenths of character. The default value depends on the configuration property [iscobol.gui.column_separation](#) whose default is 5.

Since this setting affects each column, a list of values is needed in order to determine how to draw them.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the blank space at the end of the 1st column is one cell wide and at the end of the 2nd column is 1.5. The space at the end of the other columns, if any, will be 0.5 cells, the default.

```
SEPARATION = (10, 15)
```

When a single value greater than or equal to zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Setting the property to -1 resets the list

Example - Set separations by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-gr-1, separation = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-gr-1, separation = separationAmount(columnidx)  
end-perform
```

Size

This property allows you to specify the size of the Grid control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Grid control is still computed in CELLS, but the cell size is based on the font set for the Grid control with the [Font](#) property. If no font has been defined for the Grid control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a grid with size

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    no-box  
    column-headings  
    hscroll  
    row-headings  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    cursor-frame-width 3  
    num-rows 5  
    .  
...
```

Sort-data

This property can be used to set or retrieve the sort action of each column of the grid.

It can be used in the [MSG-BEGIN-SORT](#) event as well as before issuing a sort by modifying the [Action](#) property.

When inquired it returns a string containing a sequence of numbers that describe the sort action of the columns. When modified, it expects a similar string. The format of the string is as follows:

For each column, six digits are returned. The result of sort-data for a grid with two columns will look like this: "PPPPODPPPPOD", where:

PPPP -> is the priority. A value of zero means that the column is not involved in the sort.

O -> is the ordering type. A value of 1 means ascending, a value of 2 means descending and a value of 0 means unset

D -> is the data type. It must be set to 1. Currently this is the only available value.

For example 000111000000 means that an ascending sort has been issued on the first col.

Example - Order a grid ascending on the first column and descending on the second column

```
procedure division.
...
  modify screen-1-gr-1 sort-data "000111000221"
  modify screen-1-gr-1 action action-sort
...
```

Sort-Types

This property specifies the sorting property of the Grid control columns. Multiple values can be specified enclosed between parentheses. Values are applied to cells, starting from the first. Sort types are represented by single characters.

Valid values are:

-	Not sortable. This is the default.
X	Alphanumeric sorting, case-insensitive
U, L	Alphanumeric sorting, case-sensitive
9	Numeric sorting, leading sign and current decimal-point used, all other non-digit characters ignored
I	Integer sorting, leading sign used, all other non-digit ignored
P	Positive integer sorting, all non-digit characters ignored
D, E	<p>Date sorting. Assumes three integer fields separated by one or more non-digit characters. Integers are year, month and day. Two-digit years less than 30 are treated as year 20xx while other two-digit years are treated as year 19xx.</p> <p>The default ordering of the fields is month, day, year if the current decimal point character is a point or period (.), otherwise the default ordering is day, month, year. You can explicitly specify the ordering by appending a three-character string enclosed by parenthesis, using Y, M and D to represent the year, month and day respectively. For example, D(DMY) is date order, with the date being in day-month-year order.</p> <p>Only MDY, DMY and YMD are currently valid. Invalid format strings have undefined results.</p>

Any value other than the ones described above is equivalent of "X" (alphanumeric sorting, case-insensitive).

The default sort ordering is ascending. You may specify the default order as descending by appending a caret (^) to the Sort-Types value. For example "X^" specifies a descending alphanumeric sort. Optional elements may be specified in any order. For example "D^(MDY)" and "D(MDY)^(MDY)" both specify a descending date field using the month-day-year ordering.

Ascending/descending order only determines the sort order when the user first clicks on the column. After that, the user can reverse the ordering by clicking again.

The user requests the sort by clicking on the column header. The [Column-Headings](#) style must be set in order to be sorted by the user.

The empty row produced by setting [Num-Rows](#) to "-1" is not included in the sort and is always placed at the bottom of the grid.

This property overrides the [Sortable-Columns](#) style. If both are set, the runtime behaves like if only Sort-Types was set.

Example - Define a grid where column 1 and 3 are sortable

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   column-headings  
   hscroll  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   display-columns (1, 10, 20, 30)  
   sort-types ("X", "-", "X", "-")  
   num-rows 5  
   .
```

Start-X

The only purpose of this property is to paint a rectangular region of the Grid control. See [Region-Color](#) property for a more detailed explanation.

Example - Modify a grid to set the color of a region

```
procedure division.  
...  
modify my_grid  
   start-x = 2 start-y = 2 x = 4 y = 4  
   region-color = 225  
...
```

Start-Y

The only purpose of this property is to paint a rectangular region of the Grid control. See [Region-Color](#) property for a more detailed explanation.

Example - Modify a grid to set the color of a region

```
procedure division.  
...  
modify my_grid  
    start-x = 2 start-y = 2 x = 4 y = 4  
    region-color = 225  
...
```

VPadding

This numeric property affects the height of the rows, indicating extra vertical space to be applied to each row. It is expressed as a percentage of the control's font. Its default value is 50; this makes the row's height 1.5 times the height of the font of the Grid control.

Example - Modify a grid to set the vpadding percentage

```
procedure division.  
...  
    modify screen-1-gr-1 vpadding 70  
...
```

View-Cursor-Y

When inquired, this property allows you to retrieve which row the cursor is placed in the grid represented on video. This kind of information is useful after a sort, when the order of the rows on video doesn't match with the order of the rows for the program.

Example - Inquire the View cursor Y position from a grid

```
working-storage section.  
77 ws-vc-y      pic 9(3).  
...  
procedure division.  
...  
    inquire screen-1-gr-1 view-cursor-y ws-vc-y  
...
```

View-To-Model-Y

When inquired, this property allows you to retrieve the row number for the program corresponding with the row number on video identified by the Y property. This kind of information is useful after a sort, when the order of the rows on video doesn't match with the order of the rows for the program.

Example - Get the actual row number of the fifth row you see on video

```
procedure division.  
...  
    inquire screen-1-gr-1(5, 1) view-to-model-y ws-vtm-y  
...  

```

Virtual-Width

This property is used to set the total width of the columns, regardless of the width of the Grid control.

The default value is dynamically calculated in order to maintain a width of 10 cells for the last column. The value is updated each time the user changes the width of a column. See also the [Display-Columns](#) property.

This property is used only with Grids that have the [Hscroll](#) style. In Grids where the horizontal scrolling is not available, the last column extends to the end of the Grid so that the whole Grid [Size](#) is covered by the columns.

When inquired, this property returns the width of all columns, including hidden columns.

Example - Modify a grid to set its virtual width in cells

```
procedure division.  
...  
    modify screen-1-gr-1 virtual-width 90.0  
...  

```

Visible

This property assumes a value of "0" if the Grid control is not visible, "1" if it is visible.

Example - Modify a grid to set it invisible

```
procedure division.  
...  
    modify screen-1-gr-1 visible 0  
...  

```

Vscroll-Pos

This numeric property contains the number of the topmost row currently visible.

Example - Inquire a grid to get its topmost row currently visible

```
working-storage section.  
77 ws-vs-pos    pic 9(3).  
...  
procedure division.  
...  
    inquire screen-1-gr-1 vscroll-pos ws-vs-pos  
...  

```

X

This property is used to identify a column of the Grid control. It is always used in conjunction with other properties to set or retrieve the value of a column property, or in conjunction with the [Y](#) property, to set or retrieve the value of a cell property.

Example - Set a cell-data using X and Y coordinates

```
procedure division.  
...  
  modify screen-1-gr-1 x=2 y=3 cell-data "Part 1020"  
...
```

Y

This property is used to identify a row of the Grid control. It is always used in conjunction with other properties to set or retrieve the value of a row property, or in conjunction with the [X](#) property, to set or retrieve the value of a cell property.

Example - Set a cell-data using X and Y coordinates

```
procedure division.  
...  
  modify screen-1-gr-1 x=2 y=3 cell-data="Part 1020"  
...
```

Styles

The following styles are applicable to the GRID control: [3-D](#), [Adjustable-Columns](#), [Adjustable-Rows](#), [Auto | Auto-Skip | Autoterminate](#), [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Boxed](#), [Centered-Headings](#), [Column-Headings](#), [Filterable-Columns](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Hscroll](#), [Low](#), [Lowlight](#), [No-Box](#), [No-Autosel](#), [No-Cell-Drag](#), [No-Search](#), [Paged](#), [Permanent](#), [Reordering-Columns](#), [Row-Headings](#), [Sortable-Columns](#), [Standard](#), [Temporary](#), [Tiled-Headings](#), [Use-Tab](#), [Vscroll](#), [Width-In-Cells](#).

{ 3-D | Boxed | No-Box }

3-D	The box drawn around the Grid control appears with a 3-D effect.
Boxed	A flat box is drawn around the Grid control.
No-Box	No box is drawn around the Grid control. Set this style when you need to save space.

The visual result may vary with different Swing LAF (Look And Feel).

Example - Define a boxed grid

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   hint "Grid Hint Information"  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   column-headings  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Adjustable-Columns

When this style is set, the user can change the column width by dragging the column divider with the mouse. Either the [Column-Headings](#) style or the [Num-Col-Headings](#) property must be set. When the width of a column is changed by the user, the [MSG-COL-WIDTH-CHANGED](#) event is fired.

The minimum column size is 1. The maximum is the visible portion of the grid. If the grid allows for horizontal scrolling, then the [Virtual-Width](#) property changes as the user changes column sizes.

Example - Define a grid with adjustable columns

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   column-headings  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Adjustable-Rows

When this style is set, the user can change the row height by dragging the row divider with the mouse.

This style and the property [Row-Hiding](#) cannot coexist.

Example - Define a grid with adjustable rows

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Auto | Auto-Skip | Autoterminate

This style causes the GRID to fire the MSG-FINISH-ENTRY event as soon as a cell is filled by the user. A cell is considered filled when the maximum number of digits is reached according to [Data-Types](#) property.

Example - Modify a grid to have autoterminate style

```
procedure division.  
...  
  modify screen-1-gr-1 autoterminate  
...
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a grid with low background and bold foreground

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   background-low  
   bold  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Centered-Headings

When this style is set, the content of the column headings is centered. Otherwise, the heading of each column is the same as the column itself. See the [Alignment](#) property for details.

For perfectly centered headings, columns [Separation](#) should be zero.

Example - Define a grid with centering-headings

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   background-low  
   bold  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Column-Headings

When this style is set, the first row of the Grid control is treated as a heading. The content of the column headings cannot be changed by the user. Column headings may have a different color (see the [Heading-Background-Color](#), [Heading-Color](#) and [Heading-Foreground-Color](#) properties), a different font (see the [Heading-Font](#) property) and are not scrolled. See also the [Num-Col-Headings](#) property.

Example - Define a grid with column-headings

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   background-low  
   bold  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Filterable-Columns

When this style is set, a funnel icon is shown before each column title. Clicking on that icon shows a pop-up dialog with the list of values stored in the column, a OK button and a Cancel button. You can create a filter by unchecking the values that you wish to discard and checking the values that you wish to keep. Clicking on the OK button applies the filter. Clicking on the Cancel button closes the filter dialog.

When a filter is applied, a small red cross is shown over the funnel icon. Clicking on it removes the filter and restores the original Grid data.

This style has effect only if used in conjunction with [Column-Headings](#).

Example - Define a grid with filterable columns

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   column-headings  
   filterable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Grid control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines](#) *value* CELLS".

Example - Define a grid with height and width in cells

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    height-in-cells  
    width-in-cells  
    line 5.7  
    column 3.1  
    size 34.6  
    lines 15.5  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    boxed  
    centered-headings  
    column-headings  
    adjustable-rows  
    adjustable-columns  
    hscroll  
    reordering-columns  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    heading-color 200  
    heading-divider-color 12  
    heading-font Calibri-10v0  
    cursor-frame-width 3  
    num-col-headings 1  
    num-rows 5  
    .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a grid with low background and bold foreground

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    background-low  
    bold  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    boxed  
    column-headings  
    adjustable-rows  
    adjustable-columns  
    hscroll  
    reordering-columns  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    heading-color 200  
    heading-divider-color 12  
    heading-font Calibri-10v0  
    cursor-frame-width 3  
    num-col-headings 1  
    num-rows 5  
    .
```

Hscroll

When this style is set in a Grid without [Virtual-Width](#), a horizontal scroll-bar is displayed.

The horizontal scroll-bar is always displayed if [Virtual-Width](#) is set.

Example - Define a grid with hscroll and width in cells

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   height-in-cells  
   width-in-cells  
   line 5.7  
   column 3.1  
   size 34.6  
   lines 15.5  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

No-Autosel

When this style is set, the content of the cells is not automatically highlighted when the cell editing starts and the cursor is placed at the end of the cell text.

Example - Define a grid with no-autosel style

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-autosel  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

No-Cell-Drag

When this style is set, the following events are not fired: [MSG-BEGIN-DRAG](#), [MSG-GOTO-CELL-DRAG](#) and [MSG-END-DRAG](#). This style overrides the `iscobol.gui.grid.no_cell_drag` (boolean) * configuration property.

Example - Define a grid with no-cell-drag style

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-cell-drag  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

No-Search

When this style is set, pressing CTRL+F when the focus is on the grid will not activate the integrated search feature.

Example - Define a grid where pressing CTRL+F doesn't activate the integrated search feature

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    no-search  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    boxed  
    no-cell-drag  
    centered-headings  
    column-headings  
    adjustable-rows  
    adjustable-columns  
    hscroll  
    reordering-columns  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    heading-color 200  
    heading-divider-color 12  
    cursor-frame-width 3  
    num-col-headings 1  
    num-rows 5  
    .
```

Paged

When this style is set, the Grid control is paged. It can contain only the rows that fit it. The vertical scroll-bar becomes useless and in its place four buttons are displayed.

If the [Shift] key is pressed, two of the four buttons change appearance and functionality.

Pressing the buttons, the user can access the previous or next record and the first, last, previous and next page of records.

In response to the user's clicks, the following events are fired: [MSG-PAGED-PREV](#), [MSG-PAGED-NEXT](#), [MSG-PAGED-FIRST](#), [MSG-PAGED-LAST](#), [MSG-PAGED-PREVPAGE](#), and [MSG-PAGED-NEXTPAGE](#).

See also the [File-Pos](#) property.

Example - Define a grid with paged style

```
screen section.  
...  
03 screen-1-gr-2 Grid  
   line 25.0  
   column 3.1  
   size 34.8 cells  
   lines 16.2 cells  
   id 2  
   event procedure screen-1-gr-2-evt-proc  
   no-box  
   column-headings  
   paged  
   mouse-wheel-scroll 5  
   row-dividers 1  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a grid with temporary style

```
screen section.  
...  
03 screen-1-gr-2 Grid  
   line 25.0  
   column 3.1  
   size 34.8 cells  
   lines 16.2 cells  
   id 2  
   event procedure screen-1-gr-2-evt-proc  
   no-box  
   column-headings  
   temporary  
   mouse-wheel-scroll 5  
   row-dividers 1  
   heading-font Default-Font  
   cursor-frame-width 3  
   num-rows 5  
   .  
...
```

Reordering-Columns

When this style is set, the user can change the order of the columns simply by dragging a column to its new position. No additional programming is needed because only the appearance is affected and the program still receives column information as if they were in their original positions.

Refer to the [Reordering-Col-Index](#) property for programmatically set or retrieve columns order.

Example - Define a grid with reordering-columns style

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-autosel  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   lm-on-columns 1  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Row-Headings

When this style is set, the first column of the Grid control is treated as a heading. The content of the column headings cannot be changed by the user. Row headings may have a different color (see the [Heading-Background-Color](#), [Heading-Color](#) and [Heading-Foreground-Color](#) properties), and a different font (see the [Heading-Font](#) property).

Example - Define a grid with row-headings style

```
screen section
...
03 screen-1-gr-1 Grid
  line 5.7
  column 3.1
  size 34.6 cells
  lines 15.5 cells
  id 1
  event procedure screen-1-gr-1-evt-proc
  boxed
  no-autosel
  centered-headings
  column-headings
  adjustable-rows
  adjustable-columns
  hscroll
  reordering-columns
  row-headings
  lm-on-columns 1
  row-dividers 1
  end-color 9
  heading-color 200
  cursor-frame-width 3
  num-col-headings 1
  num-rows 5
  .
...
```

Sortable-Columns

When this style is set, grid columns become sortable. When the user clicks a column heading, the grid content is sorted by that column. Hold [Ctrl] to sort by multiple columns. No additional programming is needed because only the appearance is affected and the program still receives columns information as if they were in their original positions.

The empty row produced by setting **Num-Rows** to "-1" is not included in the sort and is always placed at the bottom of the grid.

The sort logic depends on the **Data-Types** set for the column according to the following associations:

Data Types	Sort Type
X U L	Alphanumeric sort
9 I P Z	Numeric sort
D E	Date/Time sort

If the **Data-Types** is omitted, then an Alphanumeric sort is performed.

The Date/Time sort is performed according to the datetime format string. See [Data-Types for date and time](#) for details.

This style makes all the columns sortable. If you wish to have only some columns sortable, you should consider to use [Sort-Types](#) instead.

This style is overridden by the [Sort-Types](#) property. If both are set, the runtime behaves like if only Sort-Types was set.

Example - Define a grid with sortable-columns

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-autosel  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   row-headings  
   lm-on-columns 1  
   sortable-columns  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Tiled-Headings

This style causes headings to have a 3-D look.

Example - Define a grid with tiled-headings style

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-autosel  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   row-headings  
   lm-on-columns 1  
   tiled-headings  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Use-Tab

When this style is set, the user can navigate through the Grid control using the [Tab] and [Shift+Tab] keys.

Example - Define a grid that may use tab to navigate thru cells

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-autosel  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   row-headings  
   lm-on-columns 1  
   use-tab  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Vscroll

When this style is set, a vertical Scroll-Bar is displayed.

Example - Define a grid that shows a vertical scroll bar

```
screen section.  
...  
03 screen-1-gr-1 Grid  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event procedure screen-1-gr-1-evt-proc  
   boxed  
   no-autosel  
   centered-headings  
   column-headings  
   adjustable-rows  
   adjustable-columns  
   hscroll  
   reordering-columns  
   row-headings  
   lm-on-columns 1  
   vscroll  
   row-dividers 1  
   end-color 9  
   heading-color 200  
   heading-divider-color 12  
   heading-font Calibri-10v0  
   cursor-frame-width 3  
   num-col-headings 1  
   num-rows 5  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Grid control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size](#) *value* CELLS".

Example - Define a grid with height and width in cells

```
screen section.  
...  
03 screen-1-gr-1 Grid  
    height-in-cells  
    width-in-cells  
    line 5.7  
    column 3.1  
    size 34.6  
    lines 15.5  
    id 1  
    event procedure screen-1-gr-1-evt-proc  
    boxed  
    centered-headings  
    column-headings  
    adjustable-rows  
    adjustable-columns  
    hscroll  
    reordering-columns  
    lm-on-columns 1  
    row-dividers 1  
    end-color 9  
    heading-color 200  
    heading-divider-color 12  
    heading-font Calibri-10v0  
    cursor-frame-width 3  
    num-col-headings 1  
    num-rows 5  
    .
```

Events

The following events are applicable to the GRID control: CMD-GOTO, CMD-HELP, MSG-BEGIN-DRAG, MSG-BEGIN-ENTRY, MSG-BEGIN-HEADING-DRAG, MSG-BEGIN-HEADING-MENU-POPUP, MSG-BEGIN-SORT, MSG-BITMAP-CLICKED, MSG-BITMAP-DBLCLICK, MSG-CANCEL-ENTRY, MSG-COL-WIDTH-CHANGED, MSG-END-DRAG, MSG-END-HEADING-DRAG, MSG-END-MENU, MSG-FINISH-ENTRY, MSG-FINISH-SORT, MSG-GOTO-CELL, MSG-GOTO-CELL-DRAG, MSG-GOTO-CELL-MOUSE, MSG-GOTO-CELL-OUT-NEXT, MSG-GOTO-CELL-OUT-PREV, MSG-GRID-RBUTTON-DOWN, MSG-GRID-RBUTTON-UP, MSG-HEADING-CLICKED, MSG-HEADING-DBLCLICK, MSG-HEADING-DRAGGED, MSG-HEADING-MENU-POPUP, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-PAGED-FIRST, MSG-PAGED-LAST, MSG-PAGED-NEXT, MSG-PAGED-NEXTPAGE, MSG-PAGED-PREV, MSG-PAGED-PREVPAGE, MSG-ROW-HEIGHT-CHANGED, MSG-VALIDATE.

CMD-GOTO

This event is fired when the user tries to activate the Grid control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Grid control is requested. The EVENT-DATA-2 data item contains the Grid control [Help-Id](#).

MSG-BEGIN-DRAG

This event is fired when the user holds the left mouse button down and drags the mouse pointer in a Grid

control. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and the row number (respectively) of the cell where the dragging started. While dragging the [X](#), [Y](#), [Start-X](#) and [Start-Y](#) properties contain the same values as EVENT-DATA-1 and EVENT-DATA-2. This event is not fired if the user drags header cells. For this purpose see the [MSG-BEGIN-HEADING-DRAG](#) event.

MSG-BEGIN-ENTRY

This event is fired when the user starts changing the content of a cell in a Grid control. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and the row number (respectively) of the modified cell. The [X](#) and [Y](#) properties are automatically set to the value of the cell coordinates, until the user leaves that cell.

Setting EVENT-ACTION to EVENT-ACTION-FAIL allows you to protect the cell from being edited. In this way you can create read-only cells, but this method is less performing than using the [Cell-Protection](#) property.

MSG-BEGIN-HEADING-DRAG

This event is fired when the user holds the left mouse button down and drags the mouse pointer over the heading cells. The value of EVENT-DATA-1 represents the column whose heading is being dragged while EVENT-DATA-2 is always set to 1. The [X](#) and [Y](#) properties are automatically set in order to match EVENT-DATA-1 and EVENT-DATA-2. This event works with header cells. For standard cells see the [MSG-BEGIN-DRAG](#) event.

MSG-BEGIN-HEADING-MENU-POPUP

This event is fired when the user starts changing the check-box value of a menu item in a heading menu popup of a Grid control. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column number and the current value (respectively) of the check box menu item corresponding to the column.

Setting EVENT-ACTION to EVENT-ACTION-FAIL inhibits the change of the check-box menu item value.

MSG-BEGIN-SORT

This event is fired each time a sort is required for a grid with either the [Sortable-Columns](#) style or [Sort-Types](#). Depending on the sort settings, the EVENT-STATUS item has different meanings.

If the event is fired by a Grid with the [Sortable-Columns](#) style:

EVENT-DATA-1 contains a value of 0 if the sort has been issued by setting the [Action](#) property to ACTION-SORT, or 1 if the sort has been issued by clicking on the heading. EVENT-DATA-2 contains the order type:

1	no order
2	ascending
3	descending

To prevent the sort from completing and leave the grid unchanged, set EVENT-ACTION to EVENT-ACTION-FAIL. To change the sort settings, modify the [Sort-data](#) property and set EVENT-ACTION to EVENT-ACTION-COMPLETE.

If the event is fired by a Grid with [Sort-Types](#):

EVENTDATA-1 contains the column number. Setting EVENT-ACTION to EVENT-ACTION-FAIL or EVENT-ACTION-COMPLETE, the sort is not performed. Do it if you wish to provide a custom sort procedure by emptying the grid and reloading it in the desired order.

MSG-BITMAP-CLICKED

This event is fired when the user clicks on a bitmap contained in a grid cell with the left mouse button. The values of the EVENT-DATA-1 and EVENT-DATA-2 data items represent the column and the row number (respectively) of the cell containing the bitmap, as well as the [X](#) and [Y](#) properties.

MSG-BITMAP-DBLCLICK

This event is fired when the user double-clicks on a bitmap contained in a grid cell with the left mouse button. The values of the EVENT-DATA-1 and EVENT-DATA-2 data items represent the column and the row number (respectively) of the cell containing the bitmap, as well as the [X](#) and [Y](#) properties.

MSG-CANCEL-ENTRY

This event is fired when the Grid control is in entry mode and the user presses the [Escape] key to leave the cell. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell, as well as the [X](#) and [Y](#) properties. After the user has typed the [Escape] key, the cell content is restored to the original value.

MSG-COL-WIDTH-CHANGED

This event is fired when the user modifies the column size in a grid created with the [Adjustable-Columns](#) style. The EVENT-DATA-1 data item contains the column number, while EVENT-DATA-2 contains the new column width.

MSG-END-DRAG

This event is fired when the user is dragging a cell and releases the button. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell where the button has been released, as well as the [X](#) and [Y](#) properties. This event is not fired if the user drags header cells. For this purpose see the [MSG-END-HEADING-DRAG](#) event.

MSG-END-HEADING-DRAG

This event is fired when the user is dragging a header and releases the button. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell where the button has been released, as well as the [X](#) and [Y](#) properties. This event works with header cells. For standard cells see the [MSG-END-DRAG](#) event.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-FINISH-ENTRY

This event is fired when the user ends typing in a grid cell. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the edited cell, as well as the [X](#) and [Y](#) properties. Setting EVENT-ACTION to EVENT-ACTION-FAIL will keep the cell in edit mode avoiding the user to leave it.

MSG-FINISH-SORT

This event is fired when a grid with [Sort-Types](#) has completed a sorting operation. EVENT-DATA-1 contains the column number used to sort and EVENT-DATA-2 contains the record number of the last row of data in the grid. Use this to re-synchronize a data source's position for a paged grid.

MSG-GD-DBLCLICK

This event is fired when the user double clicks on a read-only protected cell in a Grid control. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and the row number (respectively) of the cell. The [X](#) and [Y](#) properties are automatically set to the value of the cell coordinates, until the user leaves that cell.

If the cell is not protected, a [MSG-BEGIN-ENTRY](#) event is fired instead.

See [Cell-Protection](#), [Column-Protection](#), [Row-Protection](#) and [Protection](#) for more information about cells protection.

MSG-GOTO-CELL

This event is fired when the user moves the cursor from a grid cell to another one using the keyboard keys. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the new cell, as well as the [X](#) and [Y](#) properties. The [Cursor-X](#) and [Cursor-Y](#) properties contain the coordinates of the cell the user has moved from. Setting EVENT-ACTION to EVENT-ACTION-FAIL will keep the cursor on the current cell instead on moving on the cell selected by the user. After it, you can move the cursor to a different cell by modifying the [Cursor-X](#) and [Cursor-Y](#) properties.

MSG-GOTO-CELL-DRAG

This event is fired each time the user moves to a different cell while dragging the cursor. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the new cell, as well as the [X](#) and [Y](#) properties. Setting EVENT-ACTION to EVENT-ACTION-FAIL will keep the cursor on the current cell instead on moving on the cell selected by the user. After it, you can move the cursor to a different cell by modifying the [Cursor-X](#) and [Cursor-Y](#) properties.

MSG-GOTO-CELL-MOUSE

This event is fired when the user moves the cursor from one grid cell to another, using the mouse. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the new cell, as well as the [X](#) and [Y](#) properties. The [Cursor-X](#) and [Cursor-Y](#) properties contain the coordinates of the cell the user has moved from. Setting EVENT-ACTION to EVENT-ACTION-FAIL will keep the cursor on the current cell instead on moving on the cell selected by the user. After it, you can move the cursor to a different cell by modifying the [Cursor-X](#) and [Cursor-Y](#) properties.

MSG-GOTO-CELL-OUT-NEXT

This event is fired when the user tries to move the cursor outside of the Grid area in one of the following conditions:

- the cursor is on the bottom-right cell and the user press
 - o TAB (only if the Grid has the [Use-Tab](#) style)
 - o the right arrow key
 - o the down arrow key (only if the Grid has not the [Paged](#) style)
 - o the Page Down key (only if the Grid has not the [Paged](#) style)
- the cursor is on a cell in the last column and the user press
 - o the right arrow key
- the cursor is on a cell in the last row and the user press
 - o the down arrow key (only if the Grid has not the [Paged](#) style)

- o the Page Down key (only if the Grid has not the [Paged](#) style)

The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the current cell.

MSG-GOTO-CELL-OUT-PREV

This event is fired when the user tries to move the cursor outside of the Grid area in one of the following conditions:

- the cursor is on the top-left cell and the user press
 - o SHIFT+TAB (only if the Grid has the [Use-Tab](#) style)
 - o the left arrow key
 - o the up arrow key (only if the Grid has not the [Paged](#) style)
 - o the Page Up key (only if the Grid has not the [Paged](#) style)
- the cursor is on a cell in the first column and the user press
 - o the left arrow key
- the cursor is on a cell in the first row and the user press
 - o the up arrow key (only if the Grid has not the [Paged](#) style)
 - o the Page Up key (only if the Grid has not the [Paged](#) style)

The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the current cell.

MSG-GRID-RBUTTON-DOWN

This event is fired when the user pushes the right mouse button and the mouse pointer is positioned on a Grid control. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell where the mouse pointer is, as well as the [X](#) and [Y](#) properties. Setting EVENT-ACTION to EVENT-ACTION-COMPLETE, avoid further processing of this event to occur, otherwise, the grid acts as if the user pressed the left mouse button.

MSG-GRID-RBUTTON-UP

This event is fired when the user releases the right mouse button and the mouse pointer is positioned on a Grid control. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell where the mouse pointer is, as well as the [X](#) and [Y](#) properties.

MSG-HEADING-CLICKED

This event is fired each time the user clicks on a cell of the grid heading. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell, as well as the [X](#) and [Y](#) properties.

MSG-HEADING-DBLCLICK

This event is fired each time the user double clicks on a cell of the grid heading. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the cell, as well as the [X](#) and [Y](#) properties.

MSG-HEADING-DRAGGED

This event is fired each time the user moves to a different cell while dragging a heading cell. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column and row number (respectively) of the new cell, as well as the *X* and *Y* properties.

MSG-HEADING-MENU-POPUP

This event is fired after a MSG-BEGIN-HEADING-MENU-POPUP event, unless such event was interrupted by setting EVENT-ACTION to EVENT-ACTION-FAIL or EVENT-ACTION-FAIL-TERMINATE. The EVENT-DATA-1 and EVENT-DATA-2 data items contain the column number and the new value (respectively) of the modified check-box menu item.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-PAGED-FIRST

This event is fired when the user presses the "First Page" button in a paged grid. The runtime responds to this event by positioning the record pointer at the beginning of the data source. Assuming that the data source is an indexed file, a START statement sets the record pointer so that a READ NEXT would retrieve the first record in the file. If EVENT-ACTION is set to EVENT-ACTION-NORMAL (default behavior), the control is emptied of data, except for any column headers and it generates a page worth of MSG-PAGED-NEXT events to fill up the first page of data.

If you plan to fill up the first page itself in response to this event, set EVENT-ACTION to EVENT-ACTION-COMPLETE to inform the grid that it should not generate the MSG-PAGED-NEXT events to fill the first page. If you cannot start at the beginning of the file (i.e. because the file is empty), set EVENT-ACTION to EVENT-ACTION-FAIL.

MSG-PAGED-LAST

This event is fired when the user presses the "Last Page" button in a paged grid. The runtime responds to this event by positioning the record pointer at the end of the data source. Assuming that the data source is an indexed file, a START statement sets the record pointer so that a READ PREVIOUS would retrieve the last record in the file. If EVENT-ACTION is set to EVENT-ACTION-NORMAL (default behavior), the control is emptied of data, except for any column headers and it generates a page worth of MSG-PAGED-PREV events to fill up the page of data.

If you plan to fill up the last page itself in response to this event, set EVENT-ACTION to EVENT-ACTION-COMPLETE to inform the grid that it should not generate the MSG-PAGED-PREV events to fill the last page. If you cannot start at the end of the file (i.e. because the file is empty), set EVENT-ACTION to EVENT-ACTION-FAIL.

MSG-PAGED-NEXT

This event is fired when the user presses the "Next Record" button in a paged grid. The expected response from the runtime is to supply the next record after the end of the grid's current data. To do this, add a new record at the end of the grid (using [Record-To-Add](#)). If the data is from an indexed file, the value of EVENT-DATA-2 is the number of READ NEXTs you need to perform to get to the appropriate record. This value is controlled by the [File-Pos](#) property. If you cannot supply the next record (i.e. because you have reached the end of the file), respond by setting EVENT-ACTION to EVENT-ACTION-FAIL.

MSG-PAGED-NEXTPAGE

This event is fired when the user presses the "Next Page" button in a paged grid. If you do not define a specific action when this event occurs, the grid implements the logic itself by generating a page worth of MSG-PAGED-NEXT events.

If you plan to fill up the first page itself in response to this event, set EVENT-ACTION to EVENT-ACTION-COMPLETE to inform the grid that it should not generate the MSG-PAGED-NEXT events to fill the next page.

MSG-PAGED-PREV

This event is fired when the user presses the "Previous Record" button in a paged grid. The expected response from the runtime is to supply the record before the first record of the grid's current data. To do this, add a new record at the start of the grid (using [Record-To-Add](#) and [Insertion-Index](#)). If the data is from an indexed file, the value of EVENT-DATA-2 is the number of READ PREVIOUS statements you need to perform to get to the appropriate record. This value is controlled by the [File-Pos](#) property. If you cannot supply the next record (i.e. because you have reached the end of the file), respond by setting EVENT-ACTION to EVENT-ACTION-FAIL.

MSG-PAGED-PREVPAGE

This event is fired when the user presses the "Previous Page" button in a paged grid. If you do not define a specific action when this event occurs, the grid implements the logic itself by generating a page worth of MSG-PAGED-PREV events.

If you plan to fill up the first page itself in response to this event, set EVENT-ACTION to EVENT-ACTION-COMPLETE to inform the grid that it should not generate the MSG-PAGED-PREV events to fill the previous page.

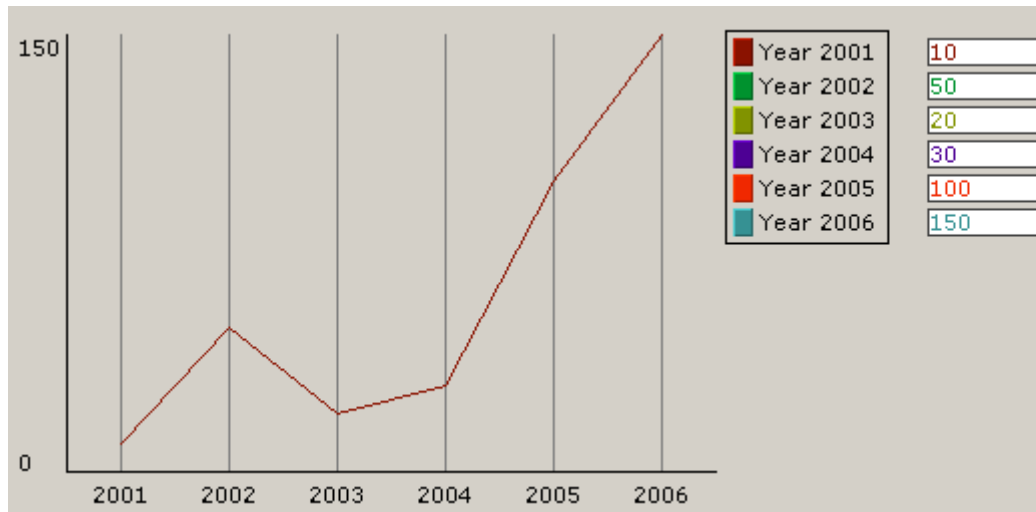
MSG-ROW-HEIGHT-CHANGED

This event is fired when the user changes the height of a row of the grid control with the style [Adjustable-Rows](#). EVENT-DATA-1 contains the number of the row whose height has been changed while EVENT-DATA-2 contains the new height.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

JAVA-BEAN



Java-beans are external components that can be used by isCOBOL as standard controls. They can be defined in screen section, used by the DISPLAY, MODIFY and INQUIRE statements. They also raise events, that should be handled in the event procedure.

Some sample programs are installed with isCOBOL. You can find them in the folder \$ISCOBOL_HOME/sample/is-java/javabeans.

Introduction to Java-Beans

Definition files

Since they usually consist of several classes, definition files are often deployed as jar files. An isCOBOL program must contain appropriate definition files to access their events and properties. A tool, called cpngen, is provided to automatically create the definition files. A separate definition file will be created for each class to which the java-bean extends, directly or indirectly.

```
cpngen [-p package] cls1 [ cls2 ... clsN] [-d outputDir]
```

Example:

To use the JCalendar bean (com.toedter.calendar.JCalendar in jcalendar.jar), you must create the definition files with one of the following commands:

```
cpngen -p com.toedter.calendar JCalendar
```

or

```
cpngen com.toedter.calendar.JCalendar
```

Since the structure of JCalendar is

```
java.lang.Object
  byjava.awt.Component
    byjava.awt.Container
      byjavax.swing.JComponent
        byjavax.swing.JPanel
          bycom.toedter.calendar.JCalendar
```

the following definition files will be created:

```
object.def
component.def
container.def
jcomponent.def
jpanel.def
jcalendar.def
```

The resulting definition files will contain event definitions

```
*> KEY event definitions, Class: java.awt.event.KeyEvent.
78 COMPONENT-KEYPRESSED VALUE 1024581858.
78 COMPONENT-KEYRELEASED VALUE 274742396.
78 COMPONENT-KEYTYPED VALUE 1303301483.
```

and the list of the available properties

```
*> Control Properties.
*> NAME: BACKGROUND , TYPE: OBJECT REFERENCE (java.awt.Color) W
*> NAME: CALENDAR , TYPE: OBJECT REFERENCE (java.util.Calendar) R/W
*> NAME: DATE , TYPE: OBJECT REFERENCE (java.util.Date) R/W
*> NAME: DAYCHOOSER , TYPE: OBJECT REFERENCE (com.toedter.calendar.JDayChooser) R
*> NAME: DECORATIONBACKGROUNDCOLOR , TYPE: OBJECT REFERENCE (java.awt.Color) R/W
*> NAME: DECORATIONBACKGROUNDVISIBLE , TYPE: NUMERIC INTEGER [VALUES 0/1] (boolean) R/W
*> NAME: DECORATIONBORDERSVISIBLE , TYPE: NUMERIC INTEGER [VALUES 0/1] (boolean) R/W
*> NAME: ENABLED , TYPE: NUMERIC INTEGER [VALUES 0/1] (boolean) R/W
*> NAME: FONT , TYPE: OBJECT REFERENCE (java.awt.Font) W
*> NAME: FOREGROUND , TYPE: OBJECT REFERENCE (java.awt.Color) W
*> NAME: LOCALE , TYPE: OBJECT REFERENCE (java.util.Locale) R/W
*> NAME: MAXDAYCHARACTERS , TYPE: NUMERIC INTEGER (int) R/W
*> NAME: MAXSELECTABLEDATE , TYPE: OBJECT REFERENCE (java.util.Date) R/W
*> NAME: MINSELECTABLEDATE , TYPE: OBJECT REFERENCE (java.util.Date) R/W
*> NAME: MONTHCHOOSER , TYPE: OBJECT REFERENCE (com.toedter.calendar.JMonthChooser) R
*> NAME: SUNDAYFOREGROUND , TYPE: OBJECT REFERENCE (java.awt.Color) R/W
*> NAME: WEEKOFYEARVISIBLE , TYPE: NUMERIC INTEGER [VALUES 0/1] (boolean) R/W
*> NAME: WEEKDAYFOREGROUND , TYPE: OBJECT REFERENCE (java.awt.Color) R/W
*> NAME: YEARCHOOSER , TYPE: OBJECT REFERENCE (com.toedter.calendar.JYearChooser) R
```

Properties

JavaBeans have common properties and special properties.

Example:

The following snippet defines a date field in the screen section:

```
03 myDateField      java-bean
  clsid             "com.toedter.calendar.JDateChooser"
  line              2
  col               2
  size              30
  lines             1.5
```

Common properties

Common properties can be set and retrieved with the standard MODIFY and INQUIRE statements, acting on the implicit control handle, myDateField in the example above.

The following statement makes the control invisible:

```
modify myDateField, visible = 0
```

Special properties

Special properties can be set or retrieved with the standard methods, setProperty and getProperty, of the java-bean object.

setProperty has two parameters: propertyName and propertyValue.

getProperty has only one parameter (propertyName) and returns propertyValue.

Property definitions are stored in the definition files created by the cpgen utility. Property definitions are comment entries with the following structure:

```
*> NAME: propertyName , TYPE: parameterType (classReference) readWriteFlag
```

propertyName is the name of the property. It must be passed, enclosed in single or double quotes, as the first parameter of the setProperty and getProperty methods. propertyName is not case-sensitive.

parameterType represents the type of the item passed as the second parameters of the setProperty method or returned by the getProperty method.

readWriteFlag allows the user to know if the property can be set, retrieved or both. Possible values are:

R	Property can be retrieved with the getProperty method.
W	Property can be set with the setProperty method.
R/W	Property can be retrieved with the getProperty method and set with the setProperty method.

The reference to the java-bean can be defined in the screen section, in the DISPLAY statement or retrieved with the INQUIRE statement.

Example:

```
03 myDateField      java-bean
  clsid             "com.toedter.calendar.JDateChooser"
  line              2
  col               2
  size             30
  lines            1.5
  object in         objDateField
```

```
display java-bean
  clsid             "com.toedter.calendar.JDateChooser"
  line              2
  col               2
  size             30
  lines            1.5
  handle in         myDateField
  object in         objDateField
```

```
inquire myDateField, object in objDateField
```

The object reference can then be used to set or retrieve special properties.

Example:

```
objDateField:>setProperty("weekOfYearVisible" 0)
```

```
SET myVariable TO objDateField:>getProperty("weekOfYearVisible")
```

Events

Each java-bean fires its own set of events. Event definitions and Event names are stored in the definition files created by the cpge utility.

Event definitions are comment entries with the following structure:

```
*>   eventClass event definitions, Class: classReference.
```

Event names are level 78 integers.

```
78 eventName VALUE eventIdValue.
```

In order to reduce the program complexity, isCOBOL provides the Event-List property, which defines the list of events to be fired. The argument of Event-List is a list of eventClass entries enclosed in parentheses. Each eventClass entry must be enclosed in single or double quotes and is not case-sensitive.

When an event is fired, event-type is set to msg-jb-event and event-data-2 is set to the event name.

Example:

The sample code has been changed, to instruct the calendar to fire the propertyChange event and handle

it in the myDateField-Event paragraph.

```
...  
working-storage section.  
...  
copy "component.def".  
...  
screen section.  
...  
03 myDateField      java-bean  
   clsid            "com.toedter.calendar.JDateChooser"  
   line             2  
   col              2  
   size             30  
   lines            1.5  
   object in        objDateField  
   event-list       ("propertyChange")  
   event procedure   myDateField-Event
```

The event procedure will look like this:

```
myDateField-Event.  
  evaluate event-type  
  when msg-jb-event  
    evaluate event-data-2  
    when COMPONENT-PROPERTYCHANGE  
      (Imperative statement)  
    end-evaluate  
  end-evaluate  
.
```

Some events may return additional information. To retrieve it, the special-names "event source" and "event object" may be defined in the working-storage section of the program.

```
...  
configuration section.  
repository.  
  class iscobol-java-bean      as "com.iscobol.gui.server.CobolGUIJavaBean"  
  class java-event-object      as "java.util.EventObject"  
...  
working-storage section.  
...  
77 eventSource is special-names event source object reference iscobol-java-bean.  
77 eventObject is special-names event object object reference java-event-object.  
...
```

No reference to a specific java-bean is made. Like other special-names, event source and event object are automatically handled by isCOBOL.

eventSource is a reference to the built-in object `com.iscobol.gui.server.CobolGUIJavaBean` and represents the object that fired the event. It provides a convenient and generic way to interact with object firing events.

eventObject is a reference to the standard java object `java.util.EventObject`, the class from which all event state objects derive. To handle a specific event fired by a java-bean, the reference to that event object must be used.

Example:

```
...
configuration section.
repository.
    class iscobol-java-bean          as "com.iscobol.gui.server.CobolGUIJavaBean"
    class java-event-object          as "java.util.EventObject"
    class property-change-event      as "java.beans.PropertyChangeEvent"
...
working-storage section.
...
77 eventSource is special-names event source object reference iscobol-java-bean.
77 eventObject is special-names event object object reference java-event-object.
77 propertyChangeEvent object reference property-change-event.
77 propertyName pic x any length.
...
myDateField-Event.
    evaluate event-type
    when msg-jb-event
        evaluate event-data-2
        when COMPONENT-PROPERTYCHANGE
            set propertyChangeEvent to eventObject as property-change-event
            set propertyName to propertyChangeEvent:>getPropertyName
            display message "The property " propertyName " has changed."
        end-evaluate
    end-evaluate
.
```

Constructors

A constructor is called each time a new instance of a class (that is, an object) is created. Constructors are used to initialize the instance variables of that object. A class, and therefore a java-bean, may have several constructors with different parameters. By default, isCOBOL uses the constructor without parameters. To use a different constructor, the Init-Params property must be specified.

```
Init-Params ( {Parameter} ... )
```

When the parameter type is ambiguous, the Init-Signature property must be specified.

```
Init-Signature "ParameterType-1 [, ParameterType-2] ..."
```

Since the Init-Params and Init-Signature determine what constructor is used to create the new instance of the java-bean, they are used only when the java-bean is being created and therefore they can be specified only in a screen item or in a DISPLAY statement, not in a MODIFY statement.

Note: These properties are parsed as standard properties by the Compiler, so they can contain only object types that can be cast to COBOL items. It's allowed to use int, long, or java.lang.String, for example, but not java.io.File, javax.swing.table.TableModel, java.util.Vector, etcetera.

Methods

Java-beans methods can be invoked with the callMethod methods of the CobolGUIJavaBean class.

```
public Object callMethod(String methodName, Object[] parameters)
```

Invokes a java-bean method

Parameters:

methodName - Name of the java-bean method to be invoked

parameters - Array of Objects representing the parameters of the method to be invoked. Use null if method has no parameters.

Returns:

The object returned by *methodName*.

```
public Object callMethod(String methodName, String signature, Object[] parameters)
```

Invokes a java-bean method with a specific signature

Parameters:

methodName - Name of the java-bean method to be invoked

signature - Signature of the method to be invoked

parameters - Array of Objects representing the parameters of the method to be invoked. To call a method with no parameters, use `callMethod(String methodName, null)`.

Returns:

The object returned by *methodName*.

```
public Object callMethod(String methodName[, Object parameter1][, Object parameter2][, Object parameter3][, Object parameter4][, Object parameter5])
```

Invokes a java-bean method

Parameters:

methodName - Name of the java-bean method to be invoked

parameters (1 to 5)- Object representing the parameters of the method to be invoked. You can use up to 5 parameters.

Returns:

The object returned by *methodName*.

Properties

The following properties are applicable to the JAVA-BEAN control: [Background-Color](#), [Bitmap-Handle](#), [Bitmap-Width](#), [Border-Color](#), [Clsid](#), [Col](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Init-Params](#), [Init-Signature](#), [Layout-data](#),

Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Object, Pop-Up Menu, Pos, Position, Size, Visible.

Background-Color

This property allows you to set or retrieve the background color of the Java-Bean control. See "[Color management](#)" for further details.

Example - Define a Java-bean control with background and foreground color

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    background-color 8  
    foreground-color 3  
    id 1  
    no-box  
    .  
...
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used for the Java-Bean.

Example - Define a Java-bean control with bitmap-handle

```
working-storage section.  
77 icon-png pic s9(9) comp-4.  
...  
screen section.  
...  
    03 screen-1-jb-1 Java-Bean  
        line 5.1  
        column 4.5  
        size 33.5 cells  
        lines 25.1 cells  
        background-color 8  
        foreground-color 3  
        id 1  
        no-box  
        bitmap-handle icon-png  
        bitmap-width 18  
        .  
...  
procedure division.  
...  
    call "w$bitmap" using wbitmap-load "icon.png" giving  
        icon-png.  
...
```

Bitmap-Width

This property defines the width in pixels of the image used.

Example - Define a Java-bean control with bitmap-handle and bitmap-width

```
working-storage section.
77 icon-png pic s9(9) comp-4.
...
screen section.
...
    03 screen-1-jb-1 Java-Bean
        line 5.1
        column 4.5
        size 33.5 cells
        lines 25.1 cells
        background-color 8
        foreground-color 3
        id 1
        no-box
        bitmap-handle icon-png
        bitmap-width 18
        .
...
procedure division.
...
    call "w$bitmap" using wbitmap-load "icon.png" giving
        icon-png.
...
```

Border-Color

This property allows you to set or retrieve the border color of the Java-Bean control. See "[Color management](#)" for further details.

Example - Define a Java-bean control with a red border

```
screen section.
...
    03 screen-1-jb-1 Java-Bean
        line 5.1
        column 4.5
        size 33.5 cells
        lines 25.1 cells
        border-color 5
        .
...
```

Clsid

This property defines the class the Java-Bean control belongs to.

Example - Define a Java-bean control with Class ID

```
screen section.  
...  
03 chart1  
   java-bean no-box  
   clsid "javax.swing.JPanel"  
   line 2  
   col 3  
   lines 16  
   size 68  
   object in JB PiePlot  
   layout-data rlm-resize-both  
   .  
...
```

[Col | Column | Pos | Position]

This property allows you to specify the Java-Bean control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Java-Bean control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Java-Bean control will be relative to the ending position of the prior Screen Section item.

When the Java-Bean control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Java-Bean, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a java-bean at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 chart1  
   java-bean no-box  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a Java-bean with css base style name, applicable with WD2

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   id 1  
   no-box  
   css-base-style-name "css-java-bean"  
   .  
...
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a Java-bean with css style name, applicable with WD2

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   id 1  
   no-box  
   css-style-name "css-java-bean"  
   .  
...
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Get the custom data from a Java-bean

```
working-storage section.  
77 ws-jb-cust-data    pic x any length.  
...  
procedure division.  
...  
    inquire screen-1-jb-1 custom-data ws-jb-cust-data  
...
```

Enabled

This property assumes a value of "0" if the Java-Bean control is disabled, "1" if it is enabled.

Example - Disable a Java-bean

```
procedure division.  
...  
    modify screen-1-jb-1 enabled 0  
...
```

Event-List

This property defines the list of events to be fired. The argument of this property is a list of eventClass entries (see Introduction above) enclosed in parentheses. Each eventClass entry must be enclosed in single or double quotes and is not case-sensitive.

COBOL events may be listed in this property as well, e.g.

```
EVENT-LIST ( "propertyChange", "mouse", CMD-GOTO )
```

To make the source more readable, you might consider to use the following equivalent syntax:

```
EVENT-LIST( "propertyChange", "mouse" )  
EVENT-LIST ( CMD-GOTO )
```

Example - Exclude CMD-GOTO COBOL event and manage propertyChange and mouse Java events

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    event-list( "propertyChange", "mouse", cmd-goto )  
    exclude-event-list 1  
    id 1  
    no-box  
    .  
...
```

Exclude-Event-List

If this property is set to "1", then none of the COBOL events in the [Event-List](#) property are fired. Other events are fired, instead. If this property is set to "0", then no event is fired at all. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Exclude CMD-GOTO COBOL event and manage propertyChange and mouse Java events

```
screen section.
...
03 screen-1-jb-1 Java-Bean
  line 5.1
  column 4.5
  size 33.5 cells
  lines 25.1 cells
  event-list( "propertyChange", "mouse", cmd-goto )
  exclude-event-list 1
  id 1
  no-box
  .
...
```

Font

This property specifies the font used to display the content of the Java-Bean control. It may be used to compute the height and the width of the Java-Bean control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a Java-bean with font property

```
working-storage section.
77 Calibri-10v0 handle of font.
...
screen section.
...
03 screen-1-jb-1 Java-Bean
  line 5.1
  column 4.5
  size 33.5 cells
  lines 25.1 cells
  background-color 8
  foreground-color 3
  font Calibri-10v0
  id 1
  no-box
  bitmap-width 18
  .
...
*> Load the font to Calibri-10v0 handle in procedure division using w$font
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Java-Bean control. See "[Color management](#)" for further details.

Example - Define a Java-bean control with background and foreground color

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   id 1  
   no-box  
   .  
...
```

Help-Id

This property allows you to assign a unique ID to the Java-Bean control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a Java-bean with help id

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   font Calibri-10v0  
   id 1  
   help-id 2020  
   no-box  
   bitmap-width 18  
   .  
...
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Java-Bean control.

Example - Define a Java-bean with hint text

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   font Calibri-10v0  
   id 1  
   hint "Hint of Java Bean"  
   no-box  
   bitmap-width 18  
   .  
...
```

Id

This property allows you to assign a unique ID to the Java-Bean control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a Java-bean with id

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   id 1  
   no-box  
   bitmap-width 18  
   .  
...
```

Init-Params

This property defines the list of parameters to be passed to the Java-Bean control constructor when it is being created.

It's useful when the Java-Bean has a particular constructor that requires some parameters. Consider the following java snippet for example:

```
JControl ctrl = new JControl(param1, param2);
```

The equivalent COBOL definition to be used in Screen Section is:

```
03 Java-Bean
...
Init-Params (param1, param2)
Object ctrl
...
.
```

Note: Since the property is parsed as a standard property by the Compiler, it can only contain object types that can be cast to COBOL items. It's permissible to use int, long, or java.lang.String, for example, but not java.io.File, javax.swing.table.TableModel, java.util.Vector, etcetera.

Init-Signature

This property defines the signature of the constructor to be used when the Java-Bean control is being created.

It's useful when the Java-Bean has more than one constructor and you wish to use one of them in particular. Consider the following JavaDoc for example:

```
JControl(int param1, int param2)
JControl(int param1, long param2)
```

The equivalent COBOL definition to be used in Screen Section in case you wish to use the first constructor is:

```
03 Java-Bean
...
Init-Signature ("int, int")
Object ctrl
...
.
```

Note: Since the property is parsed as a standard property by the Compiler, it can contain only object types that can be cast to COBOL items. It's permissible to use int, long, or java.lang.String, for example, but not java.io.File, javax.swing.table.TableModel, java.util.Vector, etcetera.

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a Java-bean with layout-data to resize in X and Y

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   id 1  
   layout-data 17  
   no-box  
   bitmap-width 18  
   .  
...
```

Line

This property allows you to specify the Java-Bean control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Java-Bean control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Java-Bean control will be relative to the starting position of the prior Screen Section item.

When the Java-Bean control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Java-Bean, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a java-bean at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Java-Bean control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Java-Bean control is still computed in CELLS, but the cell size is based on the font set for the Java-Bean

control with the [Font](#) property. If no font has been defined for the Java-Bean control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a Java-bean with height in lines

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   id 1  
   no-box  
   .  
...
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a Java-bean with maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   font Calibri-10v0  
   id 1  
   max-width 65.0  
   min-width 15.0  
   min-height 15.0  
   max-height 50.0  
   layout-data 17  
   no-box  
   .  
...
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a Java-bean with maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    background-color 8  
    foreground-color 3  
    font Calibri-10v0  
    id 1  
    max-width 65.0  
    min-width 15.0  
    min-height 15.0  
    max-height 50.0  
    layout-data 17  
    no-box  
    .  
...
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a Java-bean with maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    background-color 8  
    foreground-color 3  
    font Calibri-10v0  
    id 1  
    max-width 65.0  
    min-width 15.0  
    min-height 15.0  
    max-height 50.0  
    layout-data 17  
    no-box  
    .  
...
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a Java-bean with maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   background-color 8  
   foreground-color 3  
   font Calibri-10v0  
   id 1  
   max-width 65.0  
   min-width 15.0  
   min-height 15.0  
   max-height 50.0  
   layout-data 17  
   no-box  
   .  
...
```

Object

This property returns the reference to the Java-Bean object. The argument of this property is a data item, defined as an object reference to the class "com.iscobol.gui.server.CobolGUIJavaBean". That data item can be used to invoke the setProperty, getProperty and callMethod methods (see Introduction above).

Example - Define a Java-bean with object property

```
...  
configuration section.  
repository.  
   class BorderLayout as "java.awt.BorderLayout"  
   class JavaBean  
       as "com.iscobol.gui.server.CobolGUIJavaBean"  
...  
working-storage section.  
77 JBPiePlot    object reference JavaBean.  
...  
screen section.  
...  
03 chart1  
   java-bean no-box  
   clsid "javax.swing.JPanel"  
   line 2  
   col 3  
   lines 16  
   size 68  
   object in JBPiePlot  
   layout-data rlm-resize-both  
   .  
...
```

Pop-Up Menu

This property is ignored by Java-Bean control.

Size

This property allows you to specify the size of the Java-Bean control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Java-Bean control is still computed in CELLS, but the cell size is based on the font set for the Java-Bean control with the [Font](#) property. If no font has been defined for the Java-Bean control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a Java-bean with Size property

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   id 1  
   no-box  
   .  
...
```

Visible

This property assumes a value of "0" if the Java-Bean control is not visible, "1" if it is visible.

Example - Modify a Java-bean to make it invisible

```
procedure division.  
...  
   modify screen-1-jb-1 visible 0  
...
```

Styles

The following styles are applicable to the JAVA-BEAN control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Hscroll](#), [Low](#), [Lowlight](#), [Permanent](#), [Standard](#), [Self-Act](#), [Use-Return](#), [Use-Tab](#), [Temporary](#), [Vscroll](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a Java-bean with low background and bold foreground

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   id 1  
   no-box  
   background-low  
   bold  
   .  
...
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Java-Bean control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a Java-bean with height and width in cells styles

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5  
   lines 25.1  
   id 1  
   no-box  
   height-in-cells  
   width-in-cells  
   .  
...
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a Java-bean with low background and bold foreground

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   id 1  
   no-box  
   background-low  
   bold  
   .  
...
```

Hscroll

When this style is set, the object identified by the [Clsid](#) property is created inside a ScrollPane object.

Example - Define a Java-bean with hscroll style

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
   line 5.1  
   column 4.5  
   size 33.5 cells  
   lines 25.1 cells  
   id 1  
   no-box  
   hscroll  
   .  
...
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a Java-bean with temporary style

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    id 1  
    no-box  
    temporary  
    .  
...
```

Self-Act

When this style is set, all the events the Java-Bean control fires are trapped and no Event Procedure is started.

Example - Define a Java-bean with self-act style

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    id 1  
    no-box  
    self-act  
    .  
...
```

Use-Return

When this style is set, the user can type the [Enter] key to add a new line in a java-bean. Without it, the [Enter] key terminates the input.

Example - Define a Java-bean that uses RETURN and TAB keys without terminanting

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    id 1  
    no-box  
    use-return  
    use-tab  
    .  
...
```

Use-Tab

When this style is set, the user can use the [Tab] key to add a tab character in an java-bean. Without it, the [Tab] key is used to move from control to control.

Example - Define a Java-bean that uses RETURN and TAB keys without terminanting

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    id 1  
    no-box  
    use-return  
    use-tab  
    .  
...
```

Vscroll

When this style is set, the object identified by the `Clsid` property is created inside a ScrollPane object.

Example - Define a Java-bean that uses vscroll style

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5 cells  
    lines 25.1 cells  
    id 1  
    no-box  
    vscroll  
    .  
...
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Java-Bean control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value](#) CELLS".

Example - Define a Java-bean with height and width in cells styles

```
screen section.  
...  
03 screen-1-jb-1 Java-Bean  
    line 5.1  
    column 4.5  
    size 33.5  
    lines 25.1  
    id 1  
    no-box  
    height-in-cells  
    width-in-cells  
    .  
...
```

Events

The following events are applicable to the JAVA-BEAN control: [CMD-GOTO](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-JB-EVENT](#), [MSG-MENU-INPUT](#).

CMD-GOTO

This event is fired when the user tries to activate the JAVA-BEAN control with the mouse or by pressing the associated key letter.

With certain JAVA-BEANS that are composed of more components (e.g. the JCalendar), the mouse events that occur on the sub-components may not be correctly forwarded to the top-level component. In this case the CMD-GOTO event will be fired only if the user clicks on the top-level component.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program

should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

With certain JAVA-BEANS that are composed of more components (e.g. the JCalendar), the mouse events that occur on the sub-components may not be correctly forwarded to the top-level component. In this case the MSG-INIT-MENU event will be fired only if the user clicks on the top-level component.

MSG-JB-EVENT

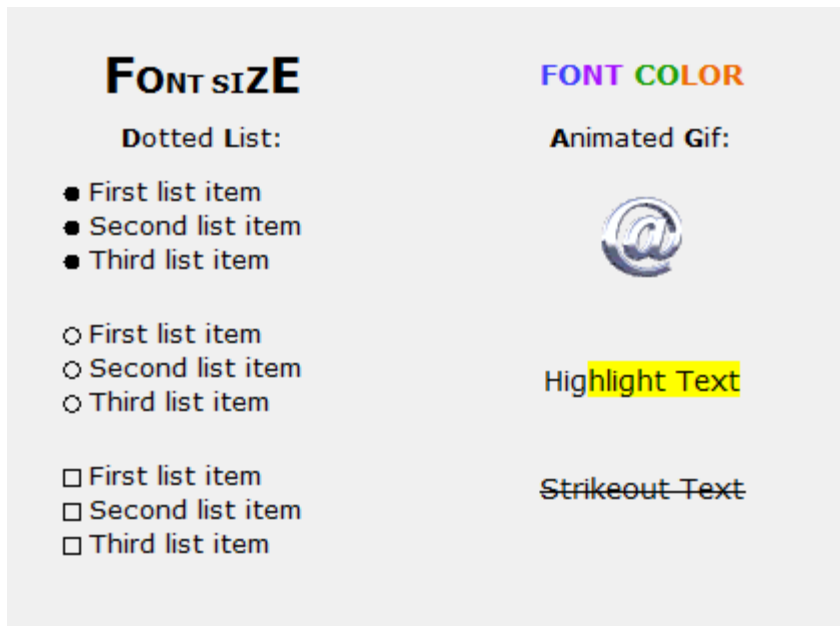
This represents a generic Java-Bean event. The EVENT-DATA-2 data item is set to a value representing the event name.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

LABEL



A Label is used to display static text on the screen. As with other read-only controls, it can render HTML code.

Properties

The following properties are applicable to the LABEL control: [Background-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Label-Offset](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Size](#), [Title](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Label control. See "[Color management](#)" for further details.

Example - Define a label with background and foreground color

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   background-color 15  
   foreground-color 6  
   id 25  
   title "This is the main Title"  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Label control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Label control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Label control will be relative to the ending position of the prior Screen Section item.

When the Label control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Label, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a label at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Label control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a label with background and foreground colors defined in the color property

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   color 519  
   id 25  
   title "This is the main Title"  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a label with css base style name, valid with EIS WD2

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   color 519  
   id 25  
   css-base-style-name "css-label-style"  
   title "This is the main Title"  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a label with css style name, valid with EIS WD2

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   color 519  
   id 25  
   css-style-name "css-label-style"  
   title "This is the main Title"  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a label

```
procedure division.  
...  
   modify screen-1-la-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Label control is disabled, "1" if it is enabled.

Example - Define a label initially disabled and enable later in procedure division

```
screen section.  
...  
  03 screen-1-la-1 Label  
    line 3.5  
    column 16.9  
    size 26.5 cells  
    lines 3.2 cells  
    color 519  
    id 25  
    enabled 0  
    title "This is the main Title"  
    .  
...  
procedure division.  
...  
  modify screen-1-la-1 enabled 1  
...
```

Font

This property specifies the font used to display the content of the Label control. It may be used to compute the height and the width of the Label control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a label with specific font

```
working-storage section.
...
77 Berlin-Sans-FB-9v0 handle of font.
...
screen section.
...
03 screen-1-la-1 Label
   line 3.5
   column 16.9
   size 26.5 cells
   lines 3.2 cells
   font Berlin-Sans-FB-9v0
   id 25
   title "This is the main Title"
   .
...
procedure division.
...
   initialize wfont-data berlin-sans-fb-9v0.
   move 9 to wfont-size.
   move "Berlin Sans FB" to wfont-name.
   set wfont-bold to false.
   set wfont-italic to false.
   set wfont-underline to false.
   set wfont-strikeout to false.
   set wfont-fixed-pitch to false.
   call "w$font" using wfont-get-font berlin-sans-fb-9v0
   wfont-data.
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Label control. See ["Color management"](#) for further details.

Example - Define a label with a foreground color

```
screen section.
...
03 screen-1-la-1 Label
   line 3.5
   column 16.9
   size 26.5 cells
   lines 3.2 cells
   background-color 15
   foreground-color 6
   id 25
   title "This is the main Title"
   .
...
```

Help-Id

This property allows you to assign a unique ID to the Label control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a label with a help-id

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   help-id 2020  
   id 25  
   title "This is the main Title"  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Label control.

Example - Define a label with a hint text

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   hint "Label for the title"  
   id 25  
   title "This is the main Title"  
   .
```

Id

This property allows you to assign a unique ID to the Label control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [isqrt.def](#).

Example - Define a label with an ID

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   title "This is the main Title"  
   .
```

Label-Offset

This property allows you to adjust the vertical position of the Label control.

The unit specified here is one hundredth of line.

The default value 20 (20 hundredths of line) makes the labels aligned to the interior text of the 3-D entry-fields.

Example - Define 2 labels on same line and column but second label will appear below given a label offset

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   title "This is the main Title"  
   .  
03 screen-1-la-2 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 26  
   label-offset 300  
   title "Sub title"  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a label that allows resize in X and Y if required by the layout manager and have limits in max and min width and height

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 5.0  
   layout-data 17  
   title "This is the main Title"  
   .
```

Line

This property allows you to specify the Label control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Label control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Label control will be relative to the starting position of the prior Screen Section item.

When the Label control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Label, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a label at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Label control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Label control is still computed in CELLS, but the cell size is based on the font set for the Label control with the [Font](#) property. If no font has been defined for the Label control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a label with dimensions in lines and size

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   title "This is the main Title"  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a label that allows resize in X and Y if required by the layout manager and have limits in max and min width and height

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 5.0  
   layout-data 17  
   title "This is the main Title"  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a label that allows resize in X and Y if required by the layout manager and have limits in

max and min width and height

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 5.0  
   layout-data 17  
   title "This is the main Title"  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a label that allows resize in X and Y if required by the layout manager and have limits in max and min width and height

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 5.0  
   layout-data 17  
   title "This is the main Title"  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a label that allows resize in X and Y if required by the layout manager and have limits in

max and min width and height

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   font Berlin-Sans-FB-9v0  
   id 25  
   max-width 30.0  
   min-width 10.0  
   min-height 2.0  
   max-height 5.0  
   layout-data 17  
   title "This is the main Title"  
   .
```

Pop-Up Menu

With this property, it is possible to associate a pop-up menu with the Label control by assigning a pop-up menu handle to it. The [LIST-BOX](#) event may be generated.

Example - Define a label with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-la-1 Label  
   line 4.3  
   column 36.7  
   size 12.4 cells  
   lines 2.9 cells  
   id 4  
   pop-up menu hmenu  
   title "R-Click Menu"  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Label control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Label control is still computed in CELLS, but the cell size is based on the font set for the Label control with the [Font](#) property. If no font has been defined for the Label control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a label with dimensions in lines and size

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   title "This is the main Title"  
   .
```

Title

The text shown in the Label control.

Example - Define a label with a title

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   title "This is the main Title"  
   .
```

Visible

This property assumes a value of "0" if the Label control is not visible, "1" if it is visible.

Example - Define a label initially not visible and make it visible on procedure division

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   title "This is the main Title"  
   visible 0  
   .  
...  
procedure division.  
...  
   modify screen-1-la-1 visible 1  
...  

```

Styles

The following styles are applicable to the LABEL control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Bottom](#), [Center](#), [Centered](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Left](#), [Low](#), [Lowlight](#), [No-Key-Letter](#), [Permanent](#), [Right](#), [Standard](#), [Temporary](#), [Top](#), [Transparent](#), [Vertical](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a label with high background

```
screen section.  
...  
03 screen-1-la-1 Label  
   background-high  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   title "This is the main Title"  
   .  
...  

```

{ Bottom | Top }

Bottom	The label title is bottom-aligned.
Top	The label title is top-aligned.

Setting one of the styles mentioned above implies that the [Vertical](#) style is set.

Example - Define a vertical label aligned to the bottom

```
screen section.  
...  
03 screen-1-la-2 Label  
   line 5.9  
   column 2.6  
   size 6.1 cells  
   lines 16.0 cells  
   font Berlin-Sans-FB-9v0  
   id 26  
   bottom  
   vertical  
   label-offset 300  
   title "Sub title"  
   .
```

{ [Center | Centered] | Left | Right }

Center, Centered	The title is centered
Left	The title is left aligned.
Right	The title is right-aligned. When used in conjunction with the Vertical style, the label is rotated 90 degrees right instead of left.

Example - Define a centered title label

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   center  
   title "This is the main Title"  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Label control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a label with height and width in cells

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5  
   lines 3.2  
   id 25  
   center  
   title "This is the main Title"  
   height-in-cells  
   width-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a label with bold foreground

```
screen section.  
...  
03 screen-1-la-1 Label  
   bold  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   title "This is the main Title"  
   .
```

No-Key-Letter

When this style is set, all "&" characters found in the title are treated as any other letter.

When this style is not set, the first "&" character in the title is not displayed; the next character can be used, in conjunction with the [Alt] key, to activate the Label control. This is the default setting.

Example - Define a label that includes the & as part of the title

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   center  
   no-key-letter  
   title "Main & Centered Title"  
   .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary label

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   id 25  
   center  
   title "Main, Centered Title"  
   temporary  
   .
```

Transparent

When this style is set, the label background becomes transparent.

Example - Define a label with white background that becomes transparent

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5 cells  
   lines 3.2 cells  
   background-color 15  
   id 25  
   center  
   transparent  
   title "Main and Centered Title"  
   .
```

Vertical

When this style is set, the label title is rotated 90 degrees to the left. When used in conjunction with the [Right](#) style, the label title is rotated 90 degrees to the right.

Example - Define a vertical label aligned to the bottom

```
screen section.  
...  
03 screen-1-la-2 Label  
   line 5.9  
   column 2.6  
   size 6.1 cells  
   lines 16.0 cells  
   font Berlin-Sans-FB-9v0  
   id 26  
   bottom  
   vertical  
   label-offset 300  
   title "Sub title"  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Label control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size](#) value CELLS".

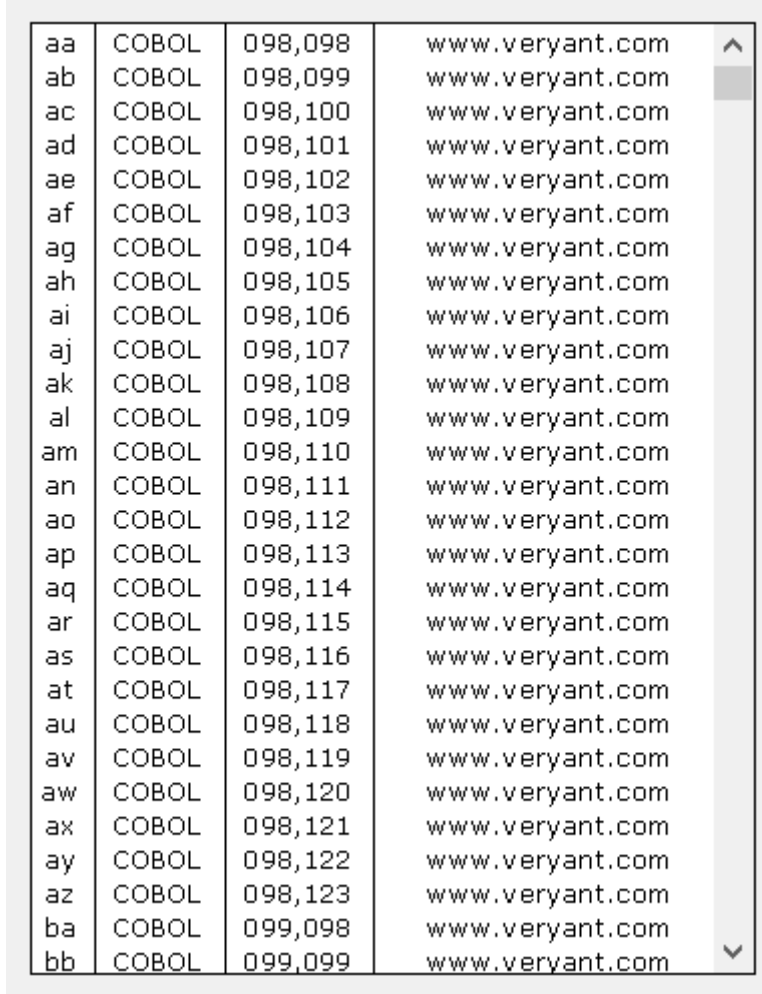
Example - Define a label with height and width in cells

```
screen section.  
...  
03 screen-1-la-1 Label  
   line 3.5  
   column 16.9  
   size 26.5  
   lines 3.2  
   id 25  
   center  
   title "This is the main Title"  
   height-in-cells  
   width-in-cells  
   .
```

Events

The Label control doesn't fire events.

LIST-BOX



aa	COBOL	098,098	www.veryant.com
ab	COBOL	098,099	www.veryant.com
ac	COBOL	098,100	www.veryant.com
ad	COBOL	098,101	www.veryant.com
ae	COBOL	098,102	www.veryant.com
af	COBOL	098,103	www.veryant.com
ag	COBOL	098,104	www.veryant.com
ah	COBOL	098,105	www.veryant.com
ai	COBOL	098,106	www.veryant.com
aj	COBOL	098,107	www.veryant.com
ak	COBOL	098,108	www.veryant.com
al	COBOL	098,109	www.veryant.com
am	COBOL	098,110	www.veryant.com
an	COBOL	098,111	www.veryant.com
ao	COBOL	098,112	www.veryant.com
ap	COBOL	098,113	www.veryant.com
aq	COBOL	098,114	www.veryant.com
ar	COBOL	098,115	www.veryant.com
as	COBOL	098,116	www.veryant.com
at	COBOL	098,117	www.veryant.com
au	COBOL	098,118	www.veryant.com
av	COBOL	098,119	www.veryant.com
aw	COBOL	098,120	www.veryant.com
ax	COBOL	098,121	www.veryant.com
ay	COBOL	098,122	www.veryant.com
az	COBOL	098,123	www.veryant.com
ba	COBOL	099,098	www.veryant.com
bb	COBOL	099,099	www.veryant.com

A List-Box is commonly used to show a list of possible choices. The user can pick one of them.

Properties

The following properties are applicable to the LIST-BOX control: [Action](#), [Alignment](#), [Background-Color](#), [Border-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Data-Columns](#), [Display-Columns](#), [Dividers](#), [Enabled](#), [Event-List](#), [Exception-Value](#), [Exclude-Event-List](#), [Export-File-Format](#), [Export-File-Name](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hidden-Data](#), [Hint](#), [Id](#), [Insertion-Index](#), [Item-To-Add](#), [Item-To-Delete](#), [Item-Value](#), [Layout-data](#), [Line](#), [Lines](#), [Lm-On-Columns](#), [Mass-Update](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Mouse-Wheel-Scroll](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Query-Index](#), [Reset-List](#), [Row-Background-Color-Pattern](#), [Row-Color-Pattern](#), [Row-Foreground-Color-Pattern](#), [Rows-Selected](#), [Search-Text](#), [Selection-Background-Color](#), [Selection-Foreground-Color](#), [Selection-Index](#), [Selection-Mode](#), [Separation](#), [Size](#), [Sort-Order](#), [Termination-Value](#), [Thumb-Position](#), [Value](#), [Visible](#).

Action

A specific action is performed when a value is assigned to this property. Only the following symbolic value,

included in the copy file `isgui.def`, can be assigned. The table below shows the action related to it:

action-copy	The content of the List-Box is copied to the clipboard. If <code>Selection-Mode</code> is set to a value greater than 0, only the selected rows are copied. Using Java7 or greater, the text format (font and colors) is copied as well.
action-export	The content of the List-Box is exported to the file name indicated by <code>Export-File-Name</code> in the format indicated by <code>Export-File-Format</code> .

Example - Modify the action property of a List-Box

```
...  
procedure division.  
...  
    modify screen-1-lb-1 action action-export  
...  

```

Alignment

This property defines the alignment for every single column of the list-box. Allowed values are:

"L"	The content of the column is left aligned. Leading spaces are ignored.
"R"	The content of the column is right aligned. Trailing spaces are ignored.
"C"	The content of the column is centered. Leading and trailing spaces are ignored.
"U"	The content of the column is left aligned. Leading spaces are kept.

Since this setting affects the alignment of each of the columns, a list of values is needed in order to determine how to align them.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the 1st column is left aligned and the 2nd column is centered. The other columns, if any, will be unaligned, the default.

```
ALIGNMENT = ("L", "C")
```

When set to space or spaces, the list is reset.

Any other single value is appended to the list. This is useful to define a user-defined appearance.

Example - Define column alignments by reading them from a Occurs

```
procedure division  
...  
modify screen-1-lb-1, alignment = spaces | Resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-lb-  
1, alignment = columnalignment(columnidx) | Sets the alignment of the next column  
end-perform
```

Background-Color

This property allows you to set or retrieve the background color of the List-Box control. See "[Color management](#)" for further details.

Example - Define a list-box with background and foreground color

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   background-color 3  
   foreground-color 7  
   id 1  
   .
```

Border-Color

This property allows you to set or retrieve the border color of the List-Box control. See "[Color management](#)" for further details. The border color is applicable only to controls with the [Boxed](#) style.

Example - define a list-box with red border

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   border-color 5  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the List-Box control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the List-Box control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the List-Box control will be relative to the ending position of the prior Screen Section item.

When the List-Box control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 List-Box, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a list-box at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 8.0  
   column 5.0  
   color 7  
   lines 10.0 cells  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the List-Box control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a list-box with background and foreground color in one property

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   color 294  
   id 1  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a list-box with css-base-style-name, applicable with WD2

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   id 1  
   css-base-style-name "css-listbox"  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a list-box with css-style-name, applicable with WD2

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    line 7.6  
    column 5.8  
    size 28.2 cells  
    lines 19.8 cells  
    id 1  
    css-style-name "css-listbox"  
    .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a list-box

```
procedure division.  
...  
    modify screen-1-lb-1 custom-data "Screen 1 custom data"  
...  

```

Data-Columns

The data in a row of the List-Box can be set or retrieved using a single alphanumeric or a group variable. This property describes the starting position of information in that variable for each column, beginning at 1. The ending position cannot be directly set: it is always the character before the first character of the next column.

We can suppose, for example, to have a three-column List-Box with the following headers: First Name, Last Name, City. The buffer necessary to add data to the list-box should be structured like this:

```
01 List-Box-Data.  
03 First_Name pic x(20).  
03 Last_Name  pic x(30).  
03 City       pic x(50).
```

The DATA-COLUMNS property should be set as follows:

```
DATA-COLUMNS = (1, 21, 51)
```

Instead of hard-coded values, it is possible to use the RECORD-POSITION syntax:

```
DATA-COLUMNS = (RECORD-POSITION OF First_Name,  
                 RECORD-POSITION OF Last_Name,  
                 RECORD-POSITION OF City)
```

This syntax avoids problems due to the modification of the item size in the buffer.

When using standard alphanumeric items, the offset of data columns is calculated in bytes, not in digits, so you should pay attention if you're using a variable length encoding (e.g. UTF-8) to store data in the List-Box record buffer.

When using national items, you can't take advantage of the RECORD-POSITION syntax. You need to use values calculated on the items length in digits, For example, for the following group item:

```
01 List-Box-Data USAGE-GROUP NATIONAL.  
03 First_Name pic N(20).  
03 Last_Name pic N(30).  
03 City pic N(50).
```

the correct DATA-COLUMNS setting is:

```
DATA-COLUMNS = (1, 21, 51)
```

and not (1, 41, 101) as RECORD-POSITION would return.

Example - Set data columns by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-lb-1, data-columns = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-lb-1, data-columns = w-data-col(columnidx)  
end-perform
```

Display-Columns

This property can be used to set or retrieve the starting position of each column in the List-Box. As a consequence, it defines the number of columns. The width of each column depends on the starting position of the next column. The last column always extends to the right side of the List-Box. If a column starts at a position that exceeds the size of the List-Box, its content is never shown. This can be useful to store information, linked to each single item, that should not be seen by the user.

Since this property must be set for each column, a list of values is needed in order to determine the starting position of each column.

When values are enclosed between parentheses, a new list is defined at once. The snippet below defines a list-box with three columns, 10 characters wide. The first column always starts at 1.

```
DISPLAY-COLUMNS = (1, 11, 21)
```

Setting this property to 0 resets the list.

When a single value greater than zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Example - Modify a list-box to set column offsets

```
procedure division.  
...  
    modify screen-1-lb-1  
        display-columns ( 1 21 29 )  
        .  
...
```

Dividers

This property defines the thickness of each line between the columns. Valid values are:

-1	Resets any value previously specified.
0	No line is drawn between the columns.
>0	A line is drawn and the number represents its thickness, in pixels.

Since this setting affects the appearance of the line between two columns, a list of values is needed in order to determine how to draw the lines.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the line between the 1st and 2nd column is one pixel wide, the line between 2nd and 3rd column is 2 pixels wide and the line between 3rd and 4th column is 3 pixels wide. The lines between the other columns, if any, will be one pixel wide, the default.

```
DIVIDERS = (1, 2, 3)
```

When a single value greater than or equal to zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Example - Set column dividers by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-lb-1, dividers = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-lb-  
1, dividers = dividerwidth(columnidx) | sets the width of the next divider  
end-perform
```

Enabled

This property assumes a value of "0" if the List-Box control is disabled, "1" if it is enabled.

Example - Enable a list-box on procedure division

```
procedure division.  
...  
    modify screen-1-lb-1 enabled 1  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a list-box with a list of events to be excluded

```
...  
03 screen-1-lb-1 List-Box  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    event-list ( cmd-goto cmd-help )  
    exclude-event-list 1  
    event procedure screen-1-lb-1-evt-proc  
    .  

```

Exception-Value

If a numeric value different from "0" is set for this property and the [Notify-Selchange](#) style is set, an exception condition for the active screen is generated when the value of the List-Box control is modified.

Example - Define a list-box with a exception-value

```
screen section  
...  
03 screen-1-lb-1 List-Box  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    exception-value 201  
    .  

```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a list-box with a list of events to be excluded

```
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   event procedure screen-1-lb-1-evt-proc  
   .
```

Export-File-Format

This property specifies the default file format in which List-Box data must be exported when the [Action](#) property is set to *action-export*.

Possible values (case insensitive) are:

Value	Meaning
XLSX	Excel Workbook
XLS	Excel 97- Excel 2003 Workbook

If the property is not set, then "XLS" is assumed.

The cell format in the generated spreadsheet is set according to the following rules:

- If the cell value contains the decimal and the group separator, the format is set to '#,##0.' plus a '0' for each digit after the decimal separator.
- If the cell value contains the decimal separator but not the group separator, the format is set to '0.' plus a '0' for each digit after the decimal separator.
- If the cell value contains the group separator but not the decimal separator, the format is set to '#,###'.
- If the cell value contains neither decimal nor group separator, no cell format is set.

Example - Define a list-box with a XLSX document associated for the export feature

```
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   export-file-format "xlsx"  
   export-file-name "output.xlsx"  
   event procedure screen-1-lb-1-evt-proc  
   .
```

Export-File-Name

This property specifies the name of the file to which List-Box data must be exported when the [Action](#) property is set to *action-export*.

If the property is not set, then "isCobolListBox.xls" is assumed.

In thin client environment, the file name is resolved on the client machine.

In WD2 environment, the file name is resolved on the web server machine.

Example - Define a list-box with a XLSX document associated for the export feature

```
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   export-file-format "xlsx"  
   export-file-name "output.xlsx"  
   event procedure screen-1-lb-1-evt-proc  
   .
```

Font

This property specifies the font used to display the content of the List-Box control. It may be used to compute the height and the width of the List-Box control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a list-box with specific Font

```
working-storage section.  
77 Calibri-10v0 handle of font.  
...  
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   font Calibri-10v0  
   id 1  
   .  
...  
*> Prior to display the screen with the list-box, load the font in  
*> procedure division using w$font  
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the List-Box control. See "[Color management](#)" for further details.

Example - Define a list-box with foreground and background color

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 7.6  
   column 5.8  
   size 28.2 cells  
   lines 19.8 cells  
   background-color 3  
   foreground-color 7  
   id 1  
   .
```

Help-Id

This property allows you to assign a unique ID to the List-Box control to be passed to the help processor. See [Help automation](#) for more information.

Example - Define a list-box with help id

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   font Calibri-10v0  
   help-id 5040  
   id 1  
   .  
...
```

Hidden-Data

This property can be used to change the hidden data of an item. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to an item.

The [Query-Index](#) property must be set first. Setting this property before the [Query-Index](#) property is not advised as it may lead to unexpected behaviors.

Example - Set hidden data for the third element

```
procedure division.  
...  
   modify screen-1-lb-1 query-index=3 hidden-data="hidden text"  
...
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the List-Box control.

Example - Define a list-box with hint text

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   font Calibri-10v0  
   help-id 5040  
   id 1  
   hint "ListBox Hint Information"  
   .  
...
```

Id

This property allows you to assign a unique ID to the List-Box control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a list-box with the ID property

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    help-id 5040  
    id 1  
    .  
...
```

Insertion-Index

This numeric property is used to set the position of the newly inserted item. The [Unsorted](#) style must be set; if not, the items are alphabetically sorted. The new item is inserted right before the item number specified here. For example, with the value "1" the next item added to the List-Box control will appear at the top of the list, while with the value "0" the item will be inserted at the end of it.

Example - Modify a list-box to insert a new item on position 5

```
procedure division.  
...  
    modify screen-1-lb-1 insertion-index 5  
    modify screen-1-lb-1 item-to-add "new item"  
...
```

Item-To-Add

When a value is assigned to this property, a new item is added to the list.

Multiple values can be added at the same time, enclosed between parentheses.

The position of the new item can be controlled with the [Insertion-Index](#) property, provided that the List-Box control has the [Unsorted](#) style set.

The value assigned to the property is split into columns according to the value set in the [Data-Columns](#) property. The values of all the columns are set simultaneously.

Example - Modify a list-box to insert a new item at the bottom of the list

```
procedure division.  
...  
  modify screen-1-lb-1 item-to-add "new item"  
...  

```

Item-To-Delete

As soon as the value of this property is modified, the corresponding item in the list is removed.

Each item is identified by a number that matches its position in the list, starting at 1.

Example - Modify a list-box to delete the second item

```
procedure division.  
...  
  modify screen-1-lb-1 item-to-delete 2  
...  

```

Item-Value

This property allows you to retrieve the value of the list-box item identified by the [Query-Index](#) property.

Example - Retrieve the text of the third item in the list

```
procedure division.  
...  
  modify screen-1-lb-1 query-index 3  
  inquire screen-1-lb-1 item-value w-value  
...  

```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a list-box that allows resize in X and Y when the layout manager requires so

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   layout-data 17  
   .  
...
```

Line

This property allows you to specify the List-Box control's vertical position. The value is specified in cells. Decimal values are allowed.

When the List-Box control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the List-Box control will be relative to the starting position of the prior Screen Section item.

When the List-Box control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 List-Box, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a list-box at line 8.0 on the screen section definition

```
...  
03 screen-1-lb-1 List-Box  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   lines 25.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the List-Box control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the List-Box control is still computed in CELLS, but the cell size is based on the font set for the List-Box control with the [Font](#) property. If no font has been defined for the List-Box control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a list-box with height in lines

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   .  
...
```

Lm-On-Columns

This property specifies the behavior of columns when the window is resized and a layout manager is involved.

(NONE)	the behavior is controlled by the <code>iscobol.gui.list.lm_on_columns (boolean)*</code> property
0	columns are not resized
1	columns are resized

Example - Define a list-box that allows resizing in columns

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   layout-data 17  
   lm-on-columns 1  
   .  
...
```

Mass-Update

Setting this property to "1" inhibits isCOBOL framework to repaint the List-Box control every time the program modifies it. This practice is recommended to increase performance when a large number of changes are applied to the List-Box control. At the end of the process it is necessary to reset the property to its default value of "0" to see the changes.

Example - Add three items in mass update mode

```
procedure division.  
...  
  modify screen-1-lb-1 mass-update 1  
  modify screen-1-lb-1 item-to-add "item 1"  
  modify screen-1-lb-1 item-to-add "item 2"  
  modify screen-1-lb-1 item-to-add "item 3"  
  modify screen-1-lb-1 mass-update 0  
...
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a list-box with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
  03 screen-1-lb-1 List-Box  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    help-id 5040  
    id 1  
    max-width 70.0  
    min-width 18.0  
    min-height 8.0  
    max-height 30.0  
    layout-data 17  
    .  
...
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a list-box with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   .  
...
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a list-box with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   .  
...
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a list-box with maximum and minimum dimensions to be used by the layout manager

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   help-id 5040  
   id 1  
   max-width 70.0  
   min-width 18.0  
   min-height 8.0  
   max-height 30.0  
   layout-data 17  
   .  
...
```

Mouse-Wheel-Scroll

This property specifies how many records must scroll in the list at each mouse wheel movement. It has effect only if the list has the [Paged](#) style.

Example - Define a list-box that scrolls three records at each mouse wheel movement

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   mouse-wheel-scroll 3  
   .  
...
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the List-Box control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a list-box with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-lb-1 List-Box  
    pop-up menu hmenu  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Query-Index

This property is used in conjunction with the [Item-Value](#) and [Hidden-Data](#) properties to retrieve the value of a specific item and to manage the hidden data bind to the item.

Each item is identified by a number that corresponds to its position in the list, starting at 1.

Example - Retrieve the text of the third item in the list

```
procedure division.  
...  
    modify screen-1-lb-1 query-index 3  
    inquire screen-1-lb-1 item-value w-value  
...
```

Reset-List

By assigning a value other than zero to this property, all the items are removed from the List-Box control.

Example - Empty a list-box

```
procedure division.  
...  
    modify screen-1-lb-1 reset-list 1  
...
```

Row-Background-Color-Pattern

This property sets the background row color pattern, as explained in the [Row-Color-Pattern](#) property. See "[Color management](#)" for further details.

Example - Modify a list-box to set its background color pattern

```
procedure division.  
...  
    modify screen-1-lb-1  
        row-background-color-pattern 7  
        row-background-color-pattern 3  
        row-background-color-pattern 1  
...
```

Row-Color-Pattern

This property sets a color pattern to be applied to the List-Box control's rows. The first color specified here is used for the first pattern row, the second color of the second pattern row and so on. The pattern is then applied to the whole control.

To obtain a "zebra" effect, add the following definition to the List-Box control:

```
ROW-COLOR-PATTERN = (257, 513)
```

Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Modify a list-box to set its background color pattern

```
procedure division.  
...  
    modify screen-1-lb-1  
        row-color-pattern 65  
        row-color-pattern 225  
        row-color-pattern 289  
...
```

Row-Foreground-Color-Pattern

This property sets the foreground row color pattern, as explained in the [Row-Color-Pattern](#) property. See "[Color management](#)" for further details.

Example - Modify a list-box to set the foreground color pattern

```
procedure division.  
...  
    modify screen-1-lb-1  
        row-foreground-color-pattern 7  
        row-foreground-color-pattern 8  
        row-foreground-color-pattern 9  
...
```

Rows-Selected

This property allows to retrieve the list of selected rows when the [Selection-Mode](#) is set to a value greater than zero.

The list of rows is returned in the format: row1 row2 ... rowN.

Example - Retrieve the list of selected rows and show it to the user

```
working-storage section.  
77 selected-rows-list pic x any length.  
  
procedure division.  
...  
    inquire screen1-lbc-1 rows-selected selected-rows-list.  
    display message selected-rows-list.  
...
```

Search-Text

This property allows you to retrieve the text the user is writing in the search area of the list-box, when the [Paged](#) style is set and the [No-Search](#) style is not set.

Example - Retrieve the text the the user is typing in the search box

```
procedure division.  
...  
    evaluate event-type  
    when NTF-PL-SEARCH  
        inquire screen-1-lb-1 search-text w-searched-text  
...
```

Selection-Background-Color

This property allows you to retrieve the background color of the currently selected row. The color is returned as an RGB value expressed by a negative number, therefore you should use signed data-items to inquire this property. See "[Color management](#)" for further details.

Example - Retrieve the selection background color

```
procedure division.  
...  
    inquire screen-1-lb-1 selection-background-color w-color  
...
```

Selection-Foreground-Color

This property allows you to retrieve the foreground color of the currently selected row. The color is returned as an RGB value expressed by a negative number, therefore you should use signed data-items to inquire this property. See "[Color management](#)" for further details.

Example - Retrieve the selection foreground color

```
procedure division.  
...  
    inquire screen-1-lb-1 selection-foreground-color w-color  
...
```

Selection-Index

This property allows you to set or retrieve the currently selected item of the list. Setting it to -1 clears any selection.

Example - Retrieve the index of the selected item

```
procedure division.  
...  
    inquire screen-1-lb-1 selection-index w-index  
...
```

Selection-Mode

This property activates the ability to select multiple rows in a List-Box.

The possible values, defined in [isgui.def](#), are:

lssm-single-selection (value 1)	It's possible to select a single row at a time. In this mode, if the Check-List style is set, radio buttons are shown before the List-Box items.
lssm-single-interval-selection (value 2)	It's possible to select more contiguous rows at a time. If the Check-List style is set, the same effect of lssm-multiple-interval-selection (value 4) is obtained.
lssm-multiple-interval-selection (value 4)	It's possible to select more rows at a time, even if they're not contiguous. In this mode, if the Check-List style is set, check boxes are shown before the List-Box items.

The list of the selected items can be retrieved by inquiring the property [Rows-Selected](#).

Example - Define a list-box checked where multiple rows can be selected:

```
screen section.  
...  
    03 screen-1-lbc-1 List-Box  
        line 5.7  
        column 3.1  
        size 34.6 cells  
        lines 15.5 cells  
        id 1  
        check-list  
        selection-mode 4  
    .
```

Separation

This property defines the amount of blank space at the end of each column, in tenths of character. The default value depends on the configuration property `iscobol.gui.column_separation` whose default is 5

Since this setting affects each column, a list of values is needed in order to determine how to draw them.

When values are enclosed between parentheses, a new list is defined at once. The snippet below specifies that the blank space at the end of the 1st column is one cell wide and at the end of the 2nd column, it is 1.5 cells wide. The space at the end of the other columns, if any, will be 0.5 cells, the default.

```
SEPARATION = (10, 15)
```

When a single value greater than or equal to zero is set, it is appended to the list. This is useful to define a user-defined appearance.

Setting the property to -1 resets the list.

Example - Set separations by reading values from a Occurs

```
procedure division.  
...  
modify screen-1-lb-1, separation = -1 | resets the list of values  
perform varying columnidx from 1 by 1 until columnidx > columncount  
    modify screen-1-lb-1, separation = separationAmount(columnidx)  
end-perform
```

Size

This property allows you to specify the size of the List-Box control. If the `PIXEL` keyword follows the value specified here, the size is computed in pixels. If either the `CELLS` keyword or the `Width-In-Cells` style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the `Width-In-Cells` style is not set, the size of the List-Box control is still computed in CELLS, but the cell size is based on the font set for the List-Box control with the `Font` property. If no font has been defined for the List-Box control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a list-box with size

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .  
...  
...
```

Sort-Order

This property affects the way items are searched in a paged list-box when the user types a text on a list-box with the [Paged](#) style set.

pl-sort-default	This is the default value, the same as pl-sort-native-ignore-case
pl-sort-none	No processing is done. The character is notified to the program, firing the NTF-PL-SEARCH event.
pl-sort-native	The items in the list are searched with a case-sensitive search. In the case of a match, the item is selected. Then, the NTF-PL-SEARCH event is fired.
pl-sort-native-ignore-case	It is the same as pl-sort-native, but the search is not case-sensitive.

Example - Define a list-box with case-sensitive search

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   sort-order pl-sort-native  
   .
```

Termination-Value

If this property is set to a numeric value different from "0" and the [Notify-Selchange](#) style is set, a termination condition for the active screen is generated when the value of the List-Box control is modified.

Example - Define a list-box with a termination-value

```
screen section  
...  
03 screen-1-lb-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   termination-value 201  
   .
```

Thumb-Position

This property allows you to set or retrieve the first visible item in a list-box. Its value is the index of the item, not the text.

Use this property only when the [Paged](#) style is not set.

Example - retrieve the text of the first visible item in the list

```
procedure division.  
...  
    inquire screen-1-lb-1 thumb-position w-index  
    modify screen-1-lb-1 query-index w-index  
    inquire screen-1-lb-1 item-value w-text  
...
```

Value

This property represents the value of the List-Box control.

When inquired, it returns the value that is currently represented.

When set, the List-Box control changes its look to represent it.

When set, the currently selected item is changed in order to match the value set.

Example - Define a list-box where the third item is selected

```
screen section.  
...  
    03 screen-1-lb-1 List-Box  
        line 5.7  
        column 3.1  
        size 34.6 cells  
        lines 15.5 cells  
        id 1  
        item-to-add ("item 1", "item 2", "item 3")  
        value "item 3"  
        .  
...
```

Visible

This property assumes a value of "0" if the List-Box control is not visible, "1" if it is visible.

Example - Modify a list-box to set it invisible

```
procedure division.  
...  
    modify screen-1-lb-1 visible 0  
...
```

Styles

The following styles are applicable to the LIST-BOX control: 3-D, Background-High, Background-Low, Background-Standard, Bold, Boxed, Check-List, Height-In-Cells, High, Highlight, Low, Lower, Lowlight, No-Box, No-Search, Notify-Dbclick, Notify-Selchange, Paged, Permanent, Standard, Temporary, Unsorted, Upper, Width-In-Cells.

{ 3-D | Boxed | No-Box }

3-D	The box drawn around the List-Box control appears with a 3-D effect.
Boxed	A flat box is drawn around the List-Box control.
No-Box	No box is drawn around the List-Box control. Set this style when you need to save space.

The visual result may vary with different Swing LAF (Look And Feel).

Example - Define a boxed list-box

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    help-id 5040  
    id 1  
    boxed  
    .
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a list-box with low background and bold foreground

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    background-low  
    bold  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```

Check-List

When this style is set, check boxes or radio buttons are automatically shown before the List-Box items,

allowing the user to easily select one or more rows.

Only one column is allowed, so [Display-Columns](#) is ignored.

The [Selection-Mode](#) property allows to switch between single selection and multiple selection.

Example - Define a list-box checked:

```
screen section.  
...  
03 screen-1-lbc-1 List-Box  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   check-list  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the List-Box control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a list-box with height and width in cells

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   height-in-cells  
   width-in-cells  
   line 5.7  
   column 3.1  
   size 34.6  
   lines 15.5  
   id 1  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a list-box with low background and bold foreground

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    background-low  
    bold  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```

{ Lower | Upper }

Lower	When this style is set, all the items are converted to lower-case characters.
Upper	When this style is set, all the items are converted to upper-case characters.

Example - Define a list-box where all items will be upper-case

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    upper  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```

No-Search

Setting this style inhibits the search function of the List-Box control.

Example - Define a paged list-box where search is disabled

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    paged  
    no-search  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```


Notify-Dblclick

This style causes a [CMD-DBLCLICK](#) event to be fired any time the user double-clicks on an item. Without this style, no event is generated under this circumstance.

Example - Define a paged list-box that fires an event upon double click

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    notify-dblclick  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```

Notify-Selchange

This style causes a [NTF-SELCHANGE](#) event to be fired any time the user selects an item. Without this style, no event is generated under this circumstance.

Example - Define a paged list-box that fires an event upon selection change

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    notify-selchange  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```

Paged

When this style is set, the List-Box control is paged. It can contain only the rows that fit it. The vertical scrollbar becomes useless and in its place four buttons are displayed.

If the [Shift] key is pressed, two of the four buttons change appearance and functionality.

Pressing the buttons, the user can access the previous or next record and the first, last, previous and next page of records.

In response to the user's clicks, the following events are fired: [NTF-PL-PREV](#), [NTF-PL-NEXT](#), [NTF-PL-FIRST](#), [NTF-PL-LAST](#), [NTF-PL-PREVPAGE](#), and [NTF-PL-NEXTPAGE](#).

Example - Define a paged list-box

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    paged  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1  
    .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary list-box

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
    temporary  
    line 5.7  
    column 3.1  
    size 34.6 cells  
    lines 15.5 cells  
    id 1.
```

Unsorted

This style causes the items contained in the list to be shown in the same order they have been added to the List-Box control.

Example - Define a list-box where items are not automatically sorted

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   unsorted  
   line 5.7  
   column 3.1  
   size 34.6 cells  
   lines 15.5 cells  
   id 1  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the List-Box control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a list-box with height and width in cells

```
screen section.  
...  
03 screen-1-lb-1 List-Box  
   height-in-cells  
   width-in-cells  
   line 5.7  
   column 3.1  
   size 34.6  
   lines 15.5  
   id 1  
   .
```

Events

The following events are applicable to the LIST-BOX control: [CMD-DBLCLICK](#), [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#), [NTF-PL-FIRST](#), [NTF-PL-LAST](#), [NTF-PL-NEXT](#), [NTF-PL-NEXTPAGE](#), [NTF-PL-PREV](#), [NTF-PL-PREVPAGE](#), [NTF-PL-SEARCH](#), [NTF-SELCHANGE](#).

CMD-DBLCLICK

This event is fired when the user double-clicks on an item of a combo or list box and either [Termination-Value](#) property or [Exception-Value](#) property is set. The EVENT-DATA-1 data item contains the index associated with the selected item.

CMD-GOTO

This event is fired when the user tries to activate the List-Box control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the List-Box control is requested. The EVENT-DATA-2 data item contains the List-Box control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

NTF-PL-FIRST

This event is fired when the user wants to scroll to the top of a PAGED list-box. The normal response is to add the first "n" records to the list where "n" is the number of lines the list box can show.

NTF-PL-LAST

This event is fired when the user wants to scroll to the bottom of a PAGED list-box. The normal response is to add the last "n" records to the list where "n" is the number of lines the list box can show.

NTF-PL-NEXT

This event is fired when the user wants to scroll one record forward in a PAGED list-box. The normal response is to add the next record in the list to the list box.

NTF-PL-NEXTPAGE

This event is fired when the user wants to scroll one page forward in a PAGED list-box. The normal response is to add the next "n" records to the list box where "n" is the number of lines the list box can show.

NTF-PL-PREV

This event is fired when the user wants to scroll one record backward in a PAGED list-box. The normal response is to add the previous record in the list to the top of the list box.

NTF-PL-PREVPAGE

This event is fired when the user wants to scroll one page backward in a PAGED list-box. The normal response is to add the previous "n" records to the top of the list box, where "n" is the number of lines the list box can

show. EVENT-DATA-1 and EVENT-DATA-2 are not used.

NTF-PL-SEARCH

This event is fired when the user types a character in the search box and the value in the search box is not currently available in the list box.

The [No-Search](#) style must not be set. The text typed can be retrieved by inquiring the [Search-Text](#) property and the EVENT-DATA-1 data item contains the length of the text.

If the [Sort-Order](#) property is set to *pl-sort-none*, the event is fired for every character inserted or removed in the search-box.

If the [Sort-Order](#) property is not set to *pl-sort-none*

- the event is fired for every character removed from the search-box
- when a character is inserted, the event is fired if the search box text doesn't match with any item in the list. Instead, if the search box text matches with an item in the list, such item is selected and the event is not fired.

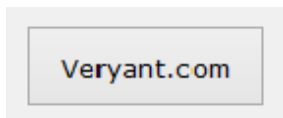
The text matches with an item when the item is equal to the text or if it starts with the text.

If the [Sort-Order](#) is set to *pl-sort-default* or *pl-sort-native-ignore-case* or the List-Box has either the [Upper](#) style or the [Lower](#) style, then the case is not considered in the comparisons. The case is considered only if the [Sort-Order](#) is *pl-sort-native* and neither the [Upper](#) style nor the [Lower](#) style is present.

NTF-SELCHANGE

This event is fired when the user selects a new item in a combo or list box created with the [Notify-Selchange](#) style. The EVENT-DATA-1 data item contains the selected item ID.

PUSH-BUTTON



A Push-Button is used to give commands to the program. It can consist of graphics, text or both.

Properties

The following properties are applicable to the PUSH-BUTTON control: [Background-Color](#), [Bitmap-Default](#), [Bitmap-Disabled](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Pressed](#), [Bitmap-Rollover](#), [Bitmap-Width](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Icon](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exception-Value](#), [Exclude-Event-List](#), [Font](#), [ForegroundColor](#), [Help-Id](#), [Hint](#), [Id](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Size](#), [Termination-Value](#), [Title](#), [Title-Position](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Push-Button control. See "[Color management](#)" for further details.

Note - Using the Windows Look and Feel, the color is not applied on the Push-Button.

Example - Define a push-button with background and foreground colors

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 12.5 cells  
   lines 4.8 cells  
   background-color 3  
   foreground-color 4  
   id 1  
   title "Accept"  
   .
```

Bitmap-Default

This property identifies the image to be displayed when the Push-Button control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

This property conflicts with [Bitmap-Number](#). If Bitmap-Default and Bitmap-Number are used together, then the first one found in the control description is considered.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-default 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

Bitmap-Disabled

This property identifies the image to be displayed when the Push-Button control is disabled. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-default 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used. The [Bitmap](#) style must be set.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-default 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

Bitmap-Number

This property identifies the image to be displayed when the Push-Button control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-number 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

Bitmap-Pressed

This property identifies the image to be displayed when the Push-Button control is pressed. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-default 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

Bitmap-Rollover

This property identifies the image to be displayed when the mouse pointer is moved over a Push-Button control. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-default 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

Bitmap-Width

This property identifies the width in pixels of the image displayed in the Push-Button control. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

Example - Define a push-button using different bitmap images for different states of it

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 70  
   lines 48  
   id 1  
   bitmap-handle icon-png  
   square  
   bitmap-default 21  
   bitmap-disabled 18  
   bitmap-rollover 19  
   bitmap-pressed 20  
   bitmap-width 18  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Push-Button control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Push-Button control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Push-Button control will be relative to the ending position of the prior Screen Section item.

When the Push-Button control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).
03 Push-Button, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a push-button at column 5.0 on the screen section definition screen section.

```
screen section.
...
03 screen-1-pb-1 Push-Button
   line 8.0
   column 5.0
   color 7
   size 45.0 cells
   id 2
   .
```

Color

This property allows you to set or retrieve the color of the Push-Button control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a push-button with background and foreground color on the same property

```
screen section.
...
03 screen-1-pb-2 Push-Button
   line 5.2
   column 18.3
   size 9.3 cells
   lines 2.4 cells
   color 164
   id 2
   title "Accept"
   .
```

Note - Using the Windows Look and Feel, the color is not applied on the Push-Button.

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a push-button with `css-base-style-name`, applicable with WD2

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   css-base-style-name "css-pb-style"  
   title "Accept"  
   .
```

Css-Icon

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

When the [Title-Position](#) property is set, both graphics and title are drawn. Otherwise, the title becomes the [Hint](#) of the control.

When this property is set, [Bitmap-Handle](#) is ignored.

Example - Define a push-button with a Font Awesome icon

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   css-icon "fa-check"  
   title "OK"  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a push-button with css-style-name, applicable with WD2

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   css-style-name "css-pb-style"  
   title "Accept"  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a push-button

```
procedure division.  
...  
   modify screen-1-br-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Push-Button control is disabled, "1" if it is enabled.

Example - Define a disabled push-button and enable it in procedure division

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   enabled 0  
   title "Accept"  
   .  
...  
procedure division.  
...  
   modify screen-1-pb-2 enabled 1  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-](#)

[Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a list of events that will be not fired for a push button

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
    line 5.2  
    column 18.3  
    size 9.3 cells  
    lines 2.4 cells  
    id 2  
    event-list ( cmd-goto msg-validate )  
    exclude-event-list 1  
    title "Accept"  
    .
```

Exception-Value

If a numeric value different from "0" is set for this property, an exception condition for the active screen is generated when the Push-Button control is clicked.

Example - Define a push-button with an exception value

```
screen section.  
...  
01 screen-1  
    exception procedure screen-1-exc-proc  
    .  
03 screen-1-pb-2 Push-Button  
    exception-value 2020  
    line 5.2  
    column 18.3  
    size 9.3 cells  
    lines 2.4 cells  
    id 2  
    event-list ( cmd-goto msg-validate )  
    exclude-event-list 1  
    title "Accept"  
    .  
...  
procedure division.  
...  
screen-1-exc-proc.  
    if key-status = 2020  
        display message "Accept button was pushed"  
    end-if  
    .
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set

to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a list of events that will be not fired for a push button

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   event-list ( cmd-goto msg-validate )  
   exclude-event-list 1  
   title "Accept"  
   .
```

Font

This property specifies the font used to display the content of the Push-Button control. It may be used to compute the height and the width of the Push-Button control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a push-button to use a particular font

```
working-storage section.  
77 Calibri-10v0 handle of font.  
...  
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   font Calibri-10v0  
   id 2  
   title "Accept"  
   .  
...  
procedure division.  
...  
   initialize wfont-data calibri-10v0.  
   move 10 to wfont-size.  
   move "Calibri" to wfont-name.  
   set wfont-bold to false.  
   set wfont-italic to false.  
   set wfont-underline to false.  
   set wfont-strikeout to false.  
   set wfont-fixed-pitch to false.  
   call "w$font" using wfont-get-font calibri-10v0 wfont-data.
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Push-Button control. See "[Color management](#)" for further details.

Note - Using the Windows Look and Feel, the color is not applied on the Push-Button.

Example - Define a push-button with background and foreground colors

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 12.5 cells  
   lines 4.8 cells  
   background-color 3  
   foreground-color 4  
   id 1  
   title "Accept"  
   .
```

Help-Id

This property allows you to assign a unique ID to the Push-Button control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a push-button with a help-id

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 12.5 cells  
   lines 4.8 cells  
   help-id 5002  
   id 1  
   title "Accept"  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Push-Button control.

Note - If the [Bitmap](#) style is set, no [Title-Position](#) has been specified and both [Title](#) and [Hint](#) have been set, then the [Title](#) text is used for the tool-tip.

Example - Define a push-button with a hint text

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 12.5 cells  
   lines 4.8 cells  
   hint "Press this button to accept changes"  
   id 1  
   title "Accept"  
   .
```

Id

This property allows you to assign a unique ID to the Push-Button control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a push-button with an ID

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 4.5  
   column 8.1  
   size 12.5 cells  
   lines 4.8 cells  
   id 1  
   title "Accept"  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a push-button with a layout-data that allows to resize in X and Y and with height and width upper and lower limits when resizing

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   max-width 60.0  
   min-width 15.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Disable"  
   .
```

Line

This property allows you to specify the Push-Button control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Push-Button control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Push-Button control will be relative to the starting position of the prior Screen Section item.

When the Push-Button control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Push-Button, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a push-button at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Push-Button control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Push-Button control is still computed in CELLS, but the cell size is based on the font set for the Push-Button control with the [Font](#) property. If no font has been defined for the Push-Button control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

When the Bitmap style is set, Lines are measured in pixels.

Example - Define a push-button with height in lines

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   title "Disable"  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a push-button with a layout-data that allows to resize in X and Y and with height and width

upper and lower limits when resizing

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   max-width 60.0  
   min-width 15.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Disable"  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a push-button with a layout-data that allows to resize in X and Y and with height and width upper and lower limits when resizing

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   max-width 60.0  
   min-width 15.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Disable"  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a push-button with a layout-data that allows to resize in X and Y and with height and width

upper and lower limits when resizing

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   max-width 60.0  
   min-width 15.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Disable"  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a push-button with a layout-data that allows to resize in X and Y and with height and width upper and lower limits when resizing

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   max-width 60.0  
   min-width 15.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Disable"  
   .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Push-Button control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a push-button with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   pop-up menu hmenu  
   line 13.7  
   column 38.1  
   size 11.7 cells  
   lines 3.6 cells  
   id 5  
   title "R-Click Menu"  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Push-Button control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Push-Button control is still computed in CELLS, but the cell size is based on the font set for the Push-Button control with the [Font](#) property. If no font has been defined for the Push-Button control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

When the **Bitmap** style is set, Size is measured in pixels.

Example - Define a push-button with the width in the size property

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   title "Disable"  
   .
```

Termination-Value

If this property is set to a numeric value different from "0", a termination condition for the active screen is generated when the Push-Button control is clicked.

Example - Define a push-button with a specific termination value

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   termination-value 2055  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   title "Disable"  
   .
```

Title

The description shown in the Push-Button control. If the [Bitmap](#) style is set, no text is shown and the title becomes the [Hint](#) of the control. If the [Title-Position](#) property is set, both text and graphics are shown.

Note - If the [Bitmap](#) style is set, no [Title-Position](#) has been specified and both [Title](#) and [Hint](#) have been set, then the [Title](#) text is used for the tool-tip.

Example - Define a push-button with a text title

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   termination-value 2055  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   title "Disable"  
   .
```

Title-Position

This property assigns the position of the title of the Push-Button control when either the [Bitmap](#) style or the [Css-Icon](#) property is set; graphics and text are combined.

Valid values usable with the [Bitmap](#) style are:

1	On the left of the image
2	On the right of the image
3	Above the image
4	Below the image
5	Over the image, centered
6	Over the image, top-left

7	Over the image, top-centered
8	Over the image, top-right
9	Over the image, center-left
10	Over the image, center-right
11	Over the image, bottom-left
12	Over the image, bottom-centered
13	Over the image, bottom-right

Valid values usable with the [Css-Icon](#) property are:

1	On the left of the image
2	On the right of the image
3	Above the image
4	Below the image

Example - Define a push-button that uses a bitmap and a text title and define the title position

```
screen section.
...
03 screen-1-pb-2 Push-Button
   line 5.2
   column 18.3
   size 93
   lines 24
   id 2
   title "Disable"
   bitmap-handle icon-png0
   bitmap-number 2
   bitmap-width 18
   title-position 2
   .
```

Visible

This property assumes a value of "0" if the Push-Button control is not visible, "1" if it is visible.

Example - Define an invisible push-button and make it visible in procedure division

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 9.3 cells  
   lines 2.4 cells  
   id 2  
   visible 0  
   title "Accept"  
   .  
...  
procedure division.  
...  
   modify screen-1-pb-2 visible 1  
...  

```

Styles

The following styles are applicable to the PUSH-BUTTON control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bitmap](#), [Bold](#), [Bottom](#), [Cancel-Button](#), [Center](#), [Default-Button](#), [Escape-Button](#), [Flat](#), [Framed](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Left](#), [Low](#), [Lowlight](#), [Multiline](#), [No-Auto-Default](#), [No-Tab](#), [Ok-Button](#), [On-Header](#), [Permanent](#), [Right](#), [Self-Act](#), [Square](#), [Standard](#), [Temporary](#), [Top](#), [Unframed](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a push-button with bold foreground and high background styles

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 93  
   lines 24  
   id 2  
   background-high  
   bold  
   title "Disable"  
   .  

```


Bitmap

The Push-Button control can contain graphics.

The [Bitmap-Handle](#) property must be set.

When the [Title-Position](#) property is set, both graphics and title are drawn. Otherwise, the title becomes the [Hint](#) of the control.

Example - Define a push-button with bitmap style

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 93  
   lines 24  
   id 2  
   bitmap  
   title "Disable"  
   bitmap-handle icon-png0  
   bitmap-number 2  
   bitmap-width 18  
   title-position 2  
   .
```

{ **Bottom** | **Center** | **Left** | **Right** | **Top** }

Bottom	The button title and icon are placed at the top of the button area.
Center	The button title and icon are horizontally centered in the button area. This is the default.
Left	The button title and icon are horizontally left aligned in the button area.
Right	The button title and icon are horizontally right aligned in the button area.
Top	The button title and icon are placed at the bottom of the button area.

The above styles have effect only if the [Multiline](#) style is not set.

Center, Left and Right can be combined with Top and Bottom as shown in the example below.

Example - Define a push-button whose title is placed in the top-left corner

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 7.4 cells  
   lines 2.5 cells  
   id 3  
   title "Button 3"  
   top left  
   .
```

{ **Cancel-Button** | **Default-Button** | **Escape-Button** | **Ok-Button** }

Cancel-Button	Set this style to create a Push-Button control suitable to cancel operations. When this style is set, the following configuration is implied: TITLE = "Cancel" EXCEPTION-VALUE = 27 ESCAPE-BUTTON
Default-Button	The Push-Button control is activated by the [Enter] key.
Escape-Button	The Push-Button control is activated by the [Esc] key.
Ok-Button	Set this style to create a Push-Button control suitable to confirm operations. When this style is set, the following configuration is implied: TITLE = "OK" TERMINATION-VALUE = 13 DEFAULT-BUTTON

Example - Define a push-button that works as a CANCEL button

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 7.4 cells  
   lines 2.5 cells  
   id 3  
   title "Cancel"  
   cancel-button  
   .
```

Flat

When this style is set, the Push-Button control has no 3-D effect. When the mouse pointer is moved over the Push-Button control, it is highlighted.

Example - Define a push-button with flat style

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 7.4 cells  
   lines 2.5 cells  
   id 3  
   flat  
   title "Save"  
   .
```

{ Framed | Unframed }

These styles have no effect, they're only supported for compatibility with other COBOLs.

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Push-Button control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a push-button with height and width in cells

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 7.4  
   lines 2.5  
   id 3  
   title "Save"  
   height-in-cells  
   width-in-cells  
   .
```

{ [Bold | High | Highlight] | [Low | Lowlight] | Standard }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a push-button with bold foreground and high background styles

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   line 5.2  
   column 18.3  
   size 93  
   lines 24  
   id 2  
   background-high  
   bold  
   title "Disable"  
   .
```

Multiline

When this style is set, the title can be displayed on multiple lines. This happens when the title does not fit the size of the Push-Button control or when it contains a LineFeed character (x"0A").

If the [Lines](#) property is set to any value, the Multiline style is implied.

Example - Define a push-button with long title to be displayed in 2 lines

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 10.7 cells  
   lines 5.3 cells  
   id 3  
   multiline  
   title "Cancel this Operation"  
   cancel-button  
   .
```

No-Auto-Default

When this style is set, the current default button is not changed.

When this style is not set, the Push-Button control becomes the default button when activated. The default button is the button the system considers depressed when the user presses the [Enter] key. Only one button can be the default for each window. This is the default setting.

Example - Define a push-button with no-auto-default style

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 10.7 cells  
   lines 5.3 cells  
   id 3  
   title "Cancel"  
   no-auto-default  
   .
```

No-Tab

Controls with this style set are skipped when the user navigates the screen using the Tab and the Backtab keys.

Example - Define a push-button that is not navigable by the tab key

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 10.7 cells  
   lines 5.3 cells  
   id 3  
   title "Go..."  
   no-tab  
   .
```

On-Header

This style affects [Bitmap](#) buttons displayed on a Ribbon. If this style is set, then the button will appear on the header line with a size of 16x16 pixels. Otherwise the button will appear on the tab-control area with the size set by [Lines](#) and [Size](#) properties.

Example - Define a push-button with bitmap and on-header styles

```
screen section.  
...  
03 screen-1-pb-2 Push-Button  
   on-header  
   line 5.2  
   column 18.3  
   size 93  
   lines 24  
   id 2  
   bitmap  
   title "Disable"  
   bitmap-handle icon-png0  
   bitmap-number 2  
   bitmap-width 18  
   title-position 2  
   .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary push-button

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 10.7 cells  
   lines 5.3 cells  
   id 3  
   title "Go..."  
   temporary  
   .
```

Self-Act

When this style is set, all the events the Push-Button control fires are trapped and no Event Procedure is started. If either the [Exception-Value](#) property or the [Termination-Value](#) property is set, the ACCEPT Statement terminates with an Exception or Termination value.

Example - Define a push-button with self-act style

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 10.7 cells  
   lines 5.3 cells  
   id 3  
   self-act  
   title "Go..."  
   .
```

Square

Treated as a comment. The compiler recognizes this style for compatibility reasons.

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Push-Button control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size](#) value CELLS".

Example - Define a push-button with width and height in cells

```
screen section.  
...  
03 screen-1-pb-3 Push-Button  
   line 12.1  
   column 15.2  
   size 7.4  
   lines 2.5  
   id 3  
   title "Save"  
   height-in-cells  
   width-in-cells  
   .
```

Events

The following events are applicable to the PUSH-BUTTON control: [CMD-CLICKED](#), [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#).

CMD-CLICKED

This event is fired when the Push-Button control is clicked. This event will always terminate the ACCEPT.

CMD-GOTO

This event is fired when the user tries to activate the Push-Button control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Push-Button control is requested. The EVENT-DATA-2 data item contains the Push-Button control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

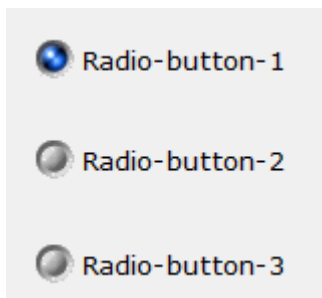
This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

RADIO-BUTTON



Radio-Buttons are always grouped and are used to select one of several available options. Since they take up considerable space, they are especially used when available options are limited, usually less than ten. When the user clicks a Radio-Button it becomes selected, while the other Radio-Buttons in the same group are unselected.

Properties

The following properties are applicable to the RADIO-BUTTON control: [Background-Color](#), [Bitmap-Default](#), [Bitmap-Disabled](#), [Bitmap-Disabled-Selected](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Pressed](#), [Bitmap-](#)

Rollover, Bitmap-Rollover-Selected, Bitmap-Selected, Bitmap-Width, Col, Color, Column, Css-Base-Style-Name, Css-Style-Name, Custom-Data, Enabled, Event-List, Exception-Value, Exclude-Event-List, Font, Foreground-Color, Group, Group-Value, Help-Id, Hint, Id, Layout-data, Left-Text-Alignment, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Termination-Value, Title, Title-Position, Value, Visible.

Background-Color

This property allows you to set or retrieve the background color of the Radio-Button control. See "[Color management](#)" for further details.

Example - Define a radio button with background and foreground color

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 9.1 cells  
   lines 4.5 cells  
   background-color 6  
   foreground-color 5  
   id 4  
   title "Option 1"  
   .
```

Bitmap-Default

This property identifies the image to be displayed when the Radio-Button control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

This property conflicts with [Bitmap-Number](#). If Bitmap-Default and Bitmap-Number are used together, then the first one found in the control description is considered.

Example - Define a radio button that uses a bitmap and defines a default image

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
    line 20.9  
    column 5.4  
    size 112  
    lines 45  
    id 4  
    title "Option 1"  
    bitmap-handle icon-png1  
    square  
    bitmap-default 1  
    bitmap-rollover 3  
    bitmap-pressed 2  
    bitmap-width 18  
    title-position 2  
    bitmap-selected 5  
    bitmap-rollover-selected 4  
    .
```

Bitmap-Disabled

This property identifies the image to be displayed when the Radio-Button control is disabled. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap and defines an image for the disabled mode

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
    line 20.9  
    column 5.4  
    size 112  
    lines 45  
    id 4  
    title "Option 1"  
    bitmap-handle icon-png1  
    square  
    bitmap-number 1  
    bitmap-disabled 6  
    bitmap-rollover 3  
    bitmap-pressed 2  
    bitmap-width 18  
    title-position 2  
    bitmap-selected 5  
    bitmap-rollover-selected 4  
    .
```

Bitmap-Disabled-Selected

This property identifies the image to be displayed when the Radio-Button control is disabled and selected. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap and defines an image for the disabled and selected mode

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   bitmap-disabled 6  
   bitmap-disabled-selected 7  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used. The [Bitmap](#) style must be set.

Example - Define a radio button that uses a bitmap and the correspondent bitmap handle

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   bitmap-disabled 6  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

Bitmap-Number

This property identifies the image to be displayed when the Radio-Button control is in normal status. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap defining the default bitmap number

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   bitmap-disabled 6  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

Bitmap-Pressed

This property identifies the image to be displayed when the Radio-Button control is selected. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap defining the image to use when the control is pressed

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   bitmap-disabled 6  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

Bitmap-Rollover

This property identifies the image to be displayed when the mouse pointer is moved over an unselected Radio-Button control. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap defining the image to use when the control is rolled over

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   bitmap-disabled 6  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

Bitmap-Rollover-Selected

This property identifies the image to be displayed when the mouse pointer is moved over a selected Radio-Button control. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap defining the image to use when the control is rolled over and selected

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
    line 20.9  
    column 5.4  
    size 112  
    lines 45  
    id 4  
    title "Option 1"  
    bitmap-handle icon-png1  
    square  
    bitmap-number 1  
    bitmap-disabled 6  
    bitmap-rollover 3  
    bitmap-pressed 2  
    bitmap-width 18  
    title-position 2  
    bitmap-selected 5  
    bitmap-rollover-selected 4  
    .
```

Bitmap-Selected

This property identifies the image to be displayed when the Radio-Button control is selected. The number corresponds to the position occupied by the image in the bitmap strip.

Example - Define a radio button that uses a bitmap defining the image to use when the control is selected

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
    line 20.9  
    column 5.4  
    size 112  
    lines 45  
    id 4  
    title "Option 1"  
    bitmap-handle icon-png1  
    square  
    bitmap-number 1  
    bitmap-disabled 6  
    bitmap-rollover 3  
    bitmap-pressed 2  
    bitmap-width 18  
    title-position 2  
    bitmap-selected 5  
    bitmap-rollover-selected 4  
    .
```

Bitmap-Width

This property identifies the width in pixels of the image displayed in the Radio-Button control. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

Example - Define a radio button that uses a bitmap defining the width of every image in the bitmap

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   bitmap-disabled 6  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Radio-Button control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Radio-Button control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Radio-Button control will be relative to the ending position of the prior Screen Section item.

When the Radio-Button control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Radio-Button, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a push-button at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-pb-1 Push-Button  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Radio-Button control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a radio button with background and foreground color on the same color property

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   color 133  
   id 5  
   title "Option 2"  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a radio button with a css base style name, applicable with WD2

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   color 133  
   id 5  
   title "Option 2"  
   css-base-style-name "css-radio-button"  
   .
```


Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a radio button with a css style name, applicable with WD2

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   color 133  
   id 5  
   title "Option 2"  
   css-style-name "css-radio-button"  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a radio-button

```
procedure division.  
...  
   modify screen-1-rb-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Radio-Button control is disabled, "1" if it is enabled.

Example - Define a radio button initially disabled and then enable it on procedure division

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   color 133  
   enabled 0  
   id 5  
   title "Option 2"  
   .  
...  
procedure division.  
...  
   modify screen-1-rb-2 enabled 1  
...
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a radio button with a list of events to be excluded

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   title "Option 2"  
   .  
...
```

Exception-Value

If a numeric value different from "0" is set for this property and the [Notify](#) style is set, an exception condition for the active screen is generated when the value of the Radio-Button control is modified.

Example - Define a radio button with an exception value to be evaluated on an exception paragraph

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   exception-value 4001  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   .
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a radio button with a list of events to be excluded

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   title "Option 2"  
   .
```

Font

This property specifies the font used to display the content of the Radio-Button control. It may be used to compute the height and the width of the Radio-Button control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a radio button with particular font

```
working-storage section.
...
77 Calibri-10v00 handle of font.
...
screen section.
...
03 screen-1-rb-2 Radio-Button
   exception-value 4001
   line 20.6
   column 21.7
   size 10.9 cells
   lines 5.4 cells
   font Calibri-10v00
   id 5
   title "Option 2"
   .
...
procedure division.
...
   initialize wfont-data calibri-10v0.
   move 10 to wfont-size.
   move "Calibri" to wfont-name.
   set wfont-bold to false.
   set wfont-italic to false.
   set wfont-underline to false.
   set wfont-strikeout to false.
   set wfont-fixed-pitch to false.
   call "w$font" using wfont-get-font calibri-10v0 wfont-data.
   move calibri-10v0 to calibri-10v00.
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Radio-Button control. See "[Color management](#)" for further details.

Example - Define a radio button with foreground and background color

```
screen section.
...
03 screen-1-rb-1 Radio-Button
   line 20.9
   column 5.4
   size 9.1 cells
   lines 4.5 cells
   background-color 6
   foreground-color 5
   id 4
   title "Option 1"
   .
```

Group

This property assigns a Radio-Button control to a group. In a group, only one Radio-Button control can be selected at a time. When the user selects a Radio-Button control, all the others in the same group are automatically unselected.

Example - Define a radio button in one group and make it the selected in the group

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   group 1  
   group-value 2  
   value 2  
   .
```

Group-Value

This property sets the value that makes a Radio-Button control selected.

When the Radio-Button control is displayed, it is selected if the value of the GROUP-VALUE property is the same as the value of the [Value](#) property.

When the user selects a Radio-Button control, the value of the [Value](#) property is set to the value of the GROUP-VALUE property.

Example - Define a radio button in one group and make it the selected in the group

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   group 1  
   group-value 2  
   value 2  
   .
```

Help-Id

This property allows you to assign a unique ID to the Radio-Button control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a radio button with a help-id

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   help-id 3020  
   id 5  
   title "Option 2"  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Radio-Button control.

Note - If the **Bitmap** style is set, no **Title-Position** has been specified and both **Title** and **Hint** have been set, then the **Title** text is used for the tool-tip.

Example - Define a radio button with a hint text

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   hint "Select this as alternative option"  
   title "Option 2"  
   .
```

Id

This property allows you to assign a unique ID to the Radio-Button control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in `iscrt.def`.

Example - Define a radio button with an ID number

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   hint "Select this as alternative option"  
   title "Option 2"  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a radio button that allows resize in X and Y and a maximum and minimum size in width and height

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   max-width 40.0  
   min-width 10.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   hint "Select this as alternative option"  
   title "Option 2"  
   .
```

Left-Text-Alignment

This property works in conjunction with the [Left-Text](#) style and controls the alignment of the text in the area on the left of the Radio-Button. Set this property to 0 to have the text right aligned or to 1 to have the text left aligned. If this property is not set, then the text is right aligned.

Example - Define a radio button with left text alignment

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   hint "Select this as alternative option"  
   title "Option 2"  
   left-text  
   left-text-alignment 1  
   .
```

Line

This property allows you to specify the Radio-Button control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Radio-Button control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Radio-Button control will be relative to the starting position of the prior Screen Section item.

When the Radio-Button control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Radio-Button, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a radio-button at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Radio-Button control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Radio-Button control is still computed in CELLS, but the cell size is based on the font set for the Radio-Button control with the [Font](#) property. If no font has been defined for the Radio-Button control, the cell size is

based on the font used for the parent window. Decimal values are allowed in this case, too.

When the Bitmap style is set, Lines are measured in pixels.

Example - Define a radio button with height in lines

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a radio button that allows resize in X and Y when the layout-manager requests it, having a maximum and minimum size

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   max-width 40.0  
   min-width 10.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Option 2"  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a radio button that allows resize in X and Y when the layout-manager requests it, having a

maximum and minimum size

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   max-width 40.0  
   min-width 10.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Option 2"  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a radio button that allows resize in X and Y when the layout-manager requests it, having a maximum and minimum size

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   max-width 40.0  
   min-width 10.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Option 2"  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a radio button that allows resize in X and Y when the layout-manager requests it, having a

maximum and minimum size

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
    line 20.6  
    column 21.7  
    size 10.9 cells  
    lines 5.4 cells  
    id 5  
    max-width 40.0  
    min-width 10.0  
    min-height 5.0  
    max-height 20.0  
    layout-data 17  
    title "Option 2"  
    .  
...
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Radio-Button control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a radio button with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
    pop-up menu hmenu  
    line 25.9  
    column 5.1  
    size 14.3 cells  
    lines 3.4 cells  
    id 6  
    title "R-Click Menu"  
    .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Radio-Button control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Radio-Button control is still computed in CELLS, but the cell size is based on the font set for the Radio-Button control with the [Font](#) property. If no font has been defined for the Radio-Button control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

When the Bitmap style is set, Size is measured in pixels.

Example - Define a radio button with the width in the size property

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   max-width 40.0  
   min-width 10.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Option 2"  
   .
```

Termination-Value

If this property is set to a numeric value different from "0" and the [Notify](#) style is set, a termination condition for the active screen is generated when the value of the Radio-Button control is modified.

Example - Define a radio button with a termination value

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   termination-value 4001  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   max-width 40.0  
   min-width 10.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   title "Option 2"  
   .
```

Title

The description shown in the Radio-Button control. If the [Bitmap](#) style is set, no text is shown and the title becomes the control hint. If the [Title-Position](#) property is set, both text and graphics are shown.

Note - If the [Bitmap](#) style is set, no [Title-Position](#) has been specified and both [Title](#) and [Hint](#) have been set, then the [Title](#) text is used for the tool-tip.

Example - Define a radio button with a title

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   .
```

Title-Position

This property assigns the position of the title of the Radio-Button control when the [Bitmap](#) style is set; graphics and text are combined. Valid values are:

1	On the left of the image
2	On the right of the image
3	Above the image
4	Below the image

Example - Define a radio button with a bitmap and title, defining the title position to the right

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   title "Option 1"  
   bitmap-handle icon-png1  
   square  
   bitmap-number 1  
   title-position 2  
   .
```

Value

This property represents the value of a group of Radio-Buttons.

When inquired, it returns the value that is currently represented in the group. For example, having these three buttons:

```
03 rb1 radio-button group 1, group-value 1.  
03 rb2 radio-button group 1, group-value 2.  
03 rb3 radio-button group 1, group-value 3.
```

if the third one is selected, any of these three statements:

```
inquire rb1 value dst-item.  
inquire rb2 value dst-item.  
inquire rb3 value dst-item.
```

will set *dst-item* to "3".

To make a Radio-Button control selected, it must be set to the value of its [Group-Value](#) property. For example, considering the above group of buttons, if you want to activate the second one, use:

```
modify rb2 value 2.
```

if you want to activate the third one, instead, use:

```
modify rb3 value 3.
```

Example - Define a radio button in one group and make it the selected in the group with the value equal to the group-value

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   group 1  
   group-value 2  
   value 2  
   .
```

Visible

This property assumes a value of "0" if the Radio-Button control is not visible, "1" if it is visible.

Example - Define a radio button initially invisible and later make it visible in the procedure division

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   title "Option 2"  
   group 1  
   group-value 2  
   value 1  
   visible 0  
   .  
...  
procedure division.  
...  
   modify screen-1-rb-2 visible 1  
...  

```

Styles

The following styles are applicable to the RADIO-BUTTON control: Background-High, Background-Low, Background-Standard, Bitmap, Bold, Flat, Framed, Height-In-Cells, High, Highlight, Left-Text, Low, Lowlight, Multiline, No-Group-Tab, No-Tab, Notify, Permanent, Self-Act, Square, Standard, Temporary, Transparent, Unframed, Vtop, Width-In-Cells.

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a radio button with High background and bold foreground

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   font Calibri-10v00  
   help-id 3020  
   id 5  
   background-high  
   bold  
   title "Option 2"  
   .
```

Bitmap

The Radio-Button control is rendered like a **Bitmap PUSH-BUTTON**. It appears pressed when the Radio-Button control is selected.

The **Bitmap-Handle** property must be set.

When the **Title-Position** property is set, both graphics and title are drawn. Otherwise, the title becomes the **Hint** of the control.

Example - Define a radio button with bitmap style

```
screen section.  
...  
03 screen-1-rb-1 Radio-Button  
   line 20.9  
   column 5.4  
   size 112  
   lines 45  
   id 4  
   bitmap  
   title "Option 1"  
   bitmap-handle icon-png1  
   bitmap-number 1  
   bitmap-rollover 3  
   bitmap-pressed 2  
   bitmap-width 18  
   title-position 2  
   group 1  
   bitmap-selected 5  
   bitmap-rollover-selected 4  
   .
```

Flat

When this style is set, the Radio-Button control has no 3-D effect. When the mouse pointer is moved over the Radio-Button control, it is highlighted.

Example - Define a radio button with flat style

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   font Calibri-10v00  
   help-id 3020  
   id 5  
   background-high  
   flat  
   title "Option 2"  
   .
```

{ Framed | Unframed }

These styles have no effect, they're only supported for compatibility with other COBOLs.

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Radio-Button control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value](#) CELLS".

Example - Define a radio button with height-in-cells style

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4  
   font Calibri-10v00  
   help-id 3020  
   id 5  
   background-high  
   height-in-cells  
   title "Option 2"  
   .
```

{ [Bold | High | Highlight] | [Low | Lowlight] | Standard }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a radio button with High background and bold foreground

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   font Calibri-10v00  
   help-id 3020  
   id 5  
   background-high  
   bold  
   title "Option 2"  
   .
```

Left-Text

When this style is set, the title is displayed on the left side. You can set the alignment of the title text through the property [Left-Text-Alignment](#).

Example - Define a radio button with left text alignment

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 5.4 cells  
   id 5  
   hint "Select this as alternative option"  
   title "Option 2"  
   left-text  
   left-text-alignment 1  
   .
```

Multiline

When this style is set, the title can be displayed on multiple lines. This happens when the title does not fit the size of the Radio-Button control or when it contains a LineFeed character (x"0A").

If the [Lines](#) property is set to any value, the Multiline style is implied.

Example - Define a radio button with multiline style to display the title on several lines if necessary

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   multiline  
   title "Alternative Option Number 2"  
   .
```

No-Group-Tab

When this style is set, the user may navigate between the radio buttons in the group by using the Tab and the Backtab keys. By default the Tab key cannot be used to navigate within individual radio buttons on the same group.

Example - Define a radio button that does not allow tab navigation in the group of radio buttons

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   no-group-tab  
   title "Alternative Option Number 2"  
   .
```

No-Tab

Controls with this style set are skipped when the user navigates the screen using the Tab and the Backtab keys.

Example - Define a radio button that does not allow tab navigation on it

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   no-tab  
   title "Alternative Option Number 2"  
   .
```

Notify

This style causes a [CMD-CLICKED](#) event to be a terminating event. The ACCEPT statement will terminate when the user activates the Radio-Button.

Example - Define a radio button that fires the cmd-clicked event when the radio button is clicked

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   notify  
   title "Alternative Option Number 2"  
   .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary radio button

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   temporary  
   title "Alternative Option Number 2"  
   .
```

Self-Act

When this style is set, all the events the Radio-Button control fires are trapped and no Event Procedure is started. If either the [Exception-Value](#) property or the [Termination-Value](#) property is set, the ACCEPT Statement terminates with an Exception or Termination value.

Example - Define a radio button with self-act style

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   self-act  
   title "Alternative Option Number 2"  
   .
```

Square

Treated as a comment. The compiler recognizes this style for compatibility reasons.

Transparent

When this style is set, the title background becomes transparent.

Example - Define a radio-button with transparent title background

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   transparent  
   title "Option 2"  
   .
```

Vtop

When this style is set, the title is aligned to the top.

When it is not set, the title is vertically centered. This is the default setting.

Example - Define a radio button with vertical alignment to the top

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9 cells  
   lines 8.1 cells  
   id 5  
   vtop  
   title "Option 2"  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Radio-Button control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a radio button with height-in-cells style

```
screen section.  
...  
03 screen-1-rb-2 Radio-Button  
   line 20.6  
   column 21.7  
   size 10.9  
   lines 5.4 cells  
   font Calibri-10v00  
   help-id 3020  
   id 5  
   background-high  
   width-in-cells  
   title "Option 2"  
   .
```

Events

The following events are applicable to the RADIO-BUTTON control: [CMD-CLICKED](#), [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#).

CMD-CLICKED

This event is fired when the Radio-Button control is clicked. If the [Notify](#) style is set, this event terminates the ACCEPT.

CMD-GOTO

This event is fired when the user tries to activate the Radio-Button control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Radio-Button control is requested. The EVENT-DATA-2 data item

contains the [Help-Id](#) for the Radio-Button control.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

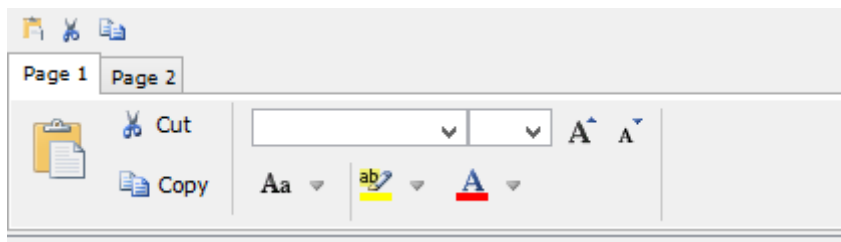
This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

RIBBON



A Ribbon is a set of controls that are placed within a tool-bar.

Ribbons have a header line followed by a Tab-Control.

The header line is used to host bitmap buttons whose size is 16x16 pixels. If no buttons are added to the header line, then the header line is not shown. In order to add a bitmap button to the header line, the ON HEADER style must be applied to the button; the runtime takes care of resizing the button icon to 16x16 if necessary.

The tab-control in the Ribbon is a Tab-Control with [Allow-Container](#) and [Multiline](#) styles and it's used to host tool-bar items, that are usually buttons.

Since controls in Ribbon cannot be activated with the keyboard, the Ribbon should contain only shortcuts to commands that can be activated elsewhere with the keyboard. Usually, it replicates menu functions.

Controls are added to the Ribbon using a Format 2 DISPLAY statement.

The following statement adds a push-button to the header line:

```
display push-button bitmap
        bitmap-handle copy-icon
        bitmap-number 1
        exception-value 101
        title "copy"
        on-header
        upon screen-1-rbn-1.
```

The following statement adds a push-button to the first page of the tab-control area:

```
display push-button bitmap
        bitmap-handle copy-icon
        bitmap-number 1
        exception-value 101
        title "copy"
        upon screen-1-rbn-1(1).
```

Properties

The following properties are applicable to the RIBBON control: [Background-Color](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), [Collapse](#), [Color](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Header-Align](#), [Hint](#), [Id](#), [Insertion-Index](#), [Layout-Manager](#), [Lines](#), [Pop-Up Menu](#), [Reset-Tabs](#), [Tab-Enabled](#), [Tab-Index](#), [Tab-Text](#), [Tab-To-Add](#), [Tab-To-Delete](#), [Value](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Ribbon control. See "[Color management](#)" for further details.

Example - Display a Ribbon control with background and foreground color

```
working-storage section.
77 screen-1-rbn-1-hdl  handle of ribbon.
...
procedure division.
...
    display ribbon
        tab-to-add ( "Page-1" "Page-2")
        bitmap-number ( 0 0)
        lines 4.4375
        background-color 8
        foreground-color 3
        id 6
        bitmap-width 16
        handle screen-1-rbn-1-hdl
        .
...

```

Bitmap-Handle

This property identifies the bitmap strip handle to be used in the Tab identified by the [Tab-Index](#) property.

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Display a Ribbon control with bitmaps on the page titles

```
working-storage section.
77 screen-1-rbn-1-hdl  handle of ribbon.
77 icon-png pic s9(9) comp-4.
...
procedure division.
...
    call "w$bitmap" using wbitmap-load "icon.png" giving
        icon-png.
...
    display ribbon
        tab-to-add ( "Page-1" "Page-2")
        bitmap-number ( 1 2)
        lines 4.4375
        color 134
        id 6
        collapse 1
        bitmap-width 18
        bitmap-handle icon-png
        handle screen-1-rbn-1-hdl
        .
...

```

Bitmap-Number

This property defines which image among the ones in the bitmap strip referenced by the [Bitmap-Handle](#) property is to be displayed in the page identified by the [Tab-Index](#) property.

When the values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Display a Ribbon control with bitmaps on the page titles

```
working-storage section.
77 screen-1-rbn-1-hdl handle of ribbon.
77 icon-png pic s9(9) comp-4.
...
procedure division.
...
    call "w$bitmap" using wbitmap-load "icon.png" giving
        icon-png.
...
    display ribbon
        tab-to-add ( "Page-1" "Page-2")
        bitmap-number ( 1 2)
        lines 4.4375
        color 134
        id 6
        collapse 1
        bitmap-width 18
        bitmap-handle icon-png
        handle screen-1-rbn-1-hdl
        .
...

```

Bitmap-Width

This property defines the width in pixels of the image displayed on the Tab identified by the [Tab-Index](#) property. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Display a Ribbon control with bitmaps on the page titles

```
working-storage section.
77 screen-1-rbn-1-hdl handle of ribbon.
77 icon-png pic s9(9) comp-4.
...
procedure division.
...
    call "w$bitmap" using wbitmap-load "icon.png" giving
        icon-png.
...
    display ribbon
        tab-to-add ( "Page-1" "Page-2")
        bitmap-number ( 1 2)
        lines 4.4375
        color 134
        id 6
        collapse 1
        bitmap-width 18
        bitmap-handle icon-png
        handle screen-1-rbn-1-hdl
        .
...

```

Collapse

This property specifies if the Ribbon tabs must to be shown collapsed or expanded. If the property is set to "1", then the tabs are collapsed. If the property is set to "0", then the tabs are expanded.

Example - Display a Ribbon control initially collapsed

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    collapse 1  
    bitmap-width 16  
    handle screen-1-rbn-1-hdl  
    .  
...
```

Color

This property allows you to set or retrieve the color of the Ribbon control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Display a Ribbon control with color property

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    bitmap-width 16  
    handle screen-1-rbn-1-hdl  
    .  
...
```

Css-Base-Style-Name

This property is ignored as the RIBBON control is not supported in a Web Direct 2.0 environment.

Css-Style-Name

This property is ignored as the RIBBON control is not supported in a Web Direct 2.0 environment.

Enabled

This property assumes a value of "0" if the Ribbon control is disabled, "1" if it is enabled.

Example - Display a Ribbon control initially disabled

```
procedure division.  
...  
  display ribbon  
  tab-to-add ( "Page-1" "Page-2")  
  bitmap-number ( 0 0)  
  lines 4.4375  
  color 134  
  enabled 0  
  id 6  
  collapse 1  
  bitmap-width 18  
  bitmap-handle icon-png  
  handle screen-1-rbn-1-hdl  
  .  
...
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Display a ribbon with excluded event list

```
procedure division.  
...  
  display ribbon  
  tab-to-add ( "Page-1" "Page-2")  
  bitmap-number ( 0 0)  
  lines 4.4375  
  color 134  
  id 6  
  bitmap-width 16  
  event-list ( cmd-tabchanged )  
  exclude-event-list 1  
  handle screen-1-rbn-1-hdl  
  .  
...
```

Exclude-Event-List

If this property is set to "1", then none of the events in the Event-List property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Display a ribbon with excluded event list

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    bitmap-width 16  
    event-list ( cmd-tabchanged )  
    exclude-event-list 1  
    handle screen-1-rbn-1-hdl  
    .  
...
```

Font

This property specifies the font that may be used to compute the height and the width of the Ribbon control. See the [Height-In-Cells](#) and [Lines](#).

Example - Display a ribbon with custom font

```
working-storage section.  
77 Calibri-10v00 handle of font.  
...  
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    bitmap-width 16  
    font Calibri-10v00  
    handle screen-1-rbn-1-hdl  
    .  
...  
*> Before displaying the screen with the toolbar, load the font to  
*> Calibri-10v00 using w$font in procedure division
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Ribbon control. See "[Color management](#)" for further details.

Example - Display a Ribbon control with background and foreground color

```
working-storage section.  
77 screen-1-rbn-1-hdl  handle of ribbon.  
...  
procedure division.  
...  
    display ribbon  
        tab-to-add ( "Page-1" "Page-2")  
        bitmap-number ( 0 0)  
        lines 4.4375  
        background-color 8  
        foreground-color 3  
        id 6  
        bitmap-width 16  
        handle screen-1-rbn-1-hdl  
        .  
...
```

Header-Align

This property specifies the alignment of the header of the RIBBON control. Possible values are:

0	Centered (default)
1	Left
2	Right

Example - Display a Ribbon control with left header align

```
procedure division.  
...  
    display ribbon  
        tab-to-add ( "Page-1" "Page-2")  
        bitmap-number ( 0 0)  
        lines 4.4375  
        color 134  
        enabled 0  
        id 6  
        collapse 1  
        header-align 1  
        bitmap-width 18  
        bitmap-handle icon-png  
        handle screen-1-rbn-1-hdl  
        .  
...
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Tab-Control control.

Example - Display a Ribbon control with Hint text

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    enabled 0  
    id 6  
    hint "Ribbon tool bar"  
    collapse 1  
    header-align 1  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...  

```

Id

This property allows you to assign a unique ID to the Ribbon control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Display a Ribbon control with Id property

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    enabled 0  
    id 6  
    hint "Ribbon tool bar"  
    collapse 1  
    header-align 1  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...  

```

Insertion-Index

This numeric property affects the position where a new Tab is added to a Ribbon control when the [Tab-To-Add](#) property is set. If this property is set to a positive value, the Tab is inserted immediately before the Tab the property refers to. When set to 0, the new Tab is appended after the last existing Tab.

Example - Modify a Ribbon control to add 1 page at position 1

```
...  
  modify screen-1-rbn-1-hdl  
    insertion-index 1 tab-to-add "Page one"  
...
```

Layout-Manager

This property associates a Layout Manager with the Ribbon. Four types of Layout Managers (defined in the [isresize.def](#) Copybook) are supported.

- LM-RESIZE
- LM-RESPONSIVE
- LM-SCALE
- LM-WRAP

Example - Display a Ribbon control with layout manager

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Ops 1" "Ops 2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    enabled 0  
    id 6  
    hint "Ribbon tool bar"  
    collapse 1  
    header-align 1  
    layout-manager lm-resize  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...
```

Lines

This property allows you to specify the height of the Ribbon control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Ribbon control is still computed in CELLS, but the cell size is based on the font set for the Ribbon control with the [Font](#) property. If no font has been defined for the Ribbon control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Display a Ribbon control with height in lines

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Ops 1" "Ops 2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    color 134  
    enabled 0  
    id 6  
    hint "Ribbon tool bar"  
    collapse 1  
    header-align 1  
    layout-manager lm-resize  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Ribbon control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Display a Ribbon control with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
procedure division.  
...  
*> Use w$menu to build the pop-up menu before displaying the Ribbon control  
...  
  display ribbon  
    tab-to-add ( "Page-1" "Page-2")  
    bitmap-number ( 0 0)  
    lines 4.4375  
    id 7  
    bitmap-width 16  
    handle screen-1-rbn-1-hdl  
    pop-up menu hmenu  
    .  
...
```

Reset-Tabs

When set to a value greater than zero, all Tabs are removed.

Example - Modify a Ribbon control to reset all its tabs

```
...procedure division.  
...  
    modify screen-1-rbn-1-hdl reset-tabs 1  
...  

```

Tab-Enabled

This property specifies whether the page identified by the [Tab-Index](#) property can be selected by the user or not. A value of 0 means that the page is disabled, so clicking on that page will produce no effects. A value of 1 means that the page is enabled, so the user can select it. Each page is enabled by default.

Example - Modify a Ribbon control to disable the third page

```
procedure division.  
...  
    modify screen-1-rbn-1-hdl tab-index 3 tab-enabled 0  
...  

```

Tab-Index

This property represents the Tab that the following properties refer to: [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), and [Tab-Text](#).

Example - Modify a Ribbon control to disable the third page

```
procedure division.  
...  
    modify screen-1-rbn-1-hdl tab-index 3 tab-enabled 0  
...  

```

Tab-Text

This property defines the title of the page identified by the [Tab-Index](#) property.

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Modify a Ribbon control to change the tab title

```
procedure division.  
...  
    modify screen-1-rbn-1-hdl  
        tab-index 3 tab-text "Ops 3 (disabled)"  
...  

```

Tab-To-Add

When this property is set, a new Tab is added to the Tab-Control control. The value of this property represents

the title of the new Tab.

Multiple Tabs can be added at the same time, by enclosing their titles between parentheses.

Example - Modify a Ribbon control to add 1 page at position 1

```
...  
  modify screen-1-rbn-1-hdl  
    insertion-index 1 tab-to-add "Page one"  
...
```

Tab-To-Delete

When set to a positive value, the corresponding Tab is removed from the Tab-Control control.

Example - Modify a Ribbon control to go to its page 2

```
procedure division.  
...  
  modify screen-1-rbn-1-hdl value 2  
...
```

Value

This property represents the value of the Ribbon control.

When inquired, it returns the value that is currently represented.

When set, the Ribbon control changes its look to represent it.

It is the currently selected Tab.

Example - Retrieve the index of the currently selected tab

```
procedure division.  
...  
  inquire screen-1-rbn-1-hdl value w-tab  
...
```

Visible

This property assumes a value of "0" if the Ribbon control is not visible, "1" if it is visible. The visibility is applied to the whole control, it's not possible to hide and show the individual pages.

Example - Modify a Ribbon control to make it invisible

```
procedure division.  
...  
  modify screen-1-rbn-1-hdl visible 0  
...
```

Styles

The following styles are applicable to the RIBBON control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Low](#), [Lowlight](#), [Permanent](#), [Relative-Offset](#), [Standard](#), [Temporary](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Display a Ribbon control with low background and bold foreground

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Ops 1" "Ops 2" "Ops 3")  
    bitmap-number ( 0 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    hint "Ribbon tool bar"  
    background-low  
    bold  
    collapse 1  
    header-align 1  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...
```

{ [Bold | High | Highlight] | [Low | Lowlight] | Standard }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Display a Ribbon control with low background and bold foreground

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Ops 1" "Ops 2" "Ops 3")  
    bitmap-number ( 0 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    hint "Ribbon tool bar"  
    background-low  
    bold  
    collapse 1  
    header-align 1  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...  

```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Ribbon control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Display a Ribbon control with height-in-cells

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Ops 1" "Ops 2" "Ops 3")  
    bitmap-number ( 0 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    hint "Ribbon tool bar"  
    height-in-cells  
    collapse 1  
    header-align 1  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...  

```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Display a Ribbon control with temporary style

```
procedure division.  
...  
    display ribbon  
        tab-to-add ( "Ops 1" "Ops 2" "Ops 3")  
        bitmap-number ( 0 0 0)  
        lines 4.4375  
        color 134  
        id 6  
        hint "Ribbon tool bar"  
        temporary  
        collapse 1  
        header-align 1  
        bitmap-width 18  
        bitmap-handle icon-png  
        handle screen-1-rbn-1-hdl  
        .  
...
```

Relative-Offset

When this style is set, the LINE and COLUMN coordinates of controls displayed on a Ribbon are relative to the Ribbon area. If the style is not set, then coordinates are relative to the window area.

Example - Display a Ribbon control with Relative-Offset style

```
procedure division.  
...  
  display ribbon  
    tab-to-add ( "Ops 1" "Ops 2" "Ops 3")  
    bitmap-number ( 0 0 0)  
    lines 4.4375  
    color 134  
    id 6  
    hint "Ribbon tool bar"  
    relative-offset  
    collapse 1  
    header-align 1  
    bitmap-width 18  
    bitmap-handle icon-png  
    handle screen-1-rbn-1-hdl  
    .  
...
```

Width-In-Cells

This style implies that the value specified for the Size property, the one setting the Ribbon control's width, is expressed in CELLS. Since the Ribbon size matches with the window size, applying this style doesn't make any difference.

Events

The following events are applicable to the RIBBON control: [CMD-TABCHANGED](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#).

CMD-TABCHANGED

This event is fired when the user selects a different Tab in the Ribbon control. The EVENT-DATA-1 data item contains the index of the Tab that has been clicked. Before activating the new Tab, the program executes the After Procedure or the Exception Procedure associated with the current control.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value.

This is used when the programmer wants to handle menu actions in the Event Procedure.

SCROLL-BAR



Scroll-Bars are rarely used because controls that need to scroll their content already have one. However, when the programmer needs to show information that cannot fit the screen or an area of it, he can use a Scroll-Bar; the user can move the Scroll-Bar slider to access information. Scroll-Bars can be horizontal or vertical.

Properties

The following properties are applicable to the SCROLL-BAR control: [Background-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Val](#), [Max-Width](#), [Min-Height](#), [Min-Val](#), [Min-Width](#), [Page-Size](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Size](#), [Value](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Scroll-Bar control. See "[Color management](#)" for further details.

Example - Define a scroll-bar with background and foreground color

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   background-color 14  
   foreground-color 4  
   id 5  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Scroll-Bar control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Scroll-Bar control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Scroll-Bar control will be relative to the ending position of the prior Screen Section item.

When the Scroll-Bar control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03  Label, COL 2, SIZE 12, (more screen options).  
03  Scroll-Bar, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a scroll-bar at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Scroll-Bar control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a scroll-bar with color property

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   id 5  
   .
```

Css-Base-Style-Name

This property is ignored as the SCROLL-BAR control is not supported in a Web Direct 2.0 environment.

Css-Style-Name

This property is ignored as the SCROLL-BAR control is not supported in a Web Direct 2.0 environment.

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to

the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a scroll-bar

```
procedure division.  
...  
    modify screen-1-sb-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property assumes a value of "0" if the Scroll-Bar control is disabled, "1" if it is enabled.

Example - Modify a scroll-bar to disable it

```
procedure division.  
...  
    modify screen-1-sb-1 enabled 0  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a scroll-bar with excluded event list

```
screen section.  
...  
    03 screen-1-sb-1 Scroll-Bar  
        line 4.0  
        column 41.5  
        size 2.9 cells  
        lines 39.7 cells  
        color 229  
        id 5  
        event-list ( cmd-goto cmd-help )  
        exclude-event-list 1  
        .  
...  

```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a scroll-bar with excluded event list

```
screen section.  
...  
    03 screen-1-sb-1 Scroll-Bar  
        line 4.0  
        column 41.5  
        size 2.9 cells  
        lines 39.7 cells  
        color 229  
        id 5  
        event-list ( cmd-goto cmd-help )  
        exclude-event-list 1  
        .  
...
```

Font

This property specifies the font that may be used to compute the height and the width of the Scroll-Bar control. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a scroll-bar with font for sizing reference

```
working-storage section.  
77 Calibri-10v00 handle of font.  
...  
screen section.  
...  
    03 screen-1-sb-1 Scroll-Bar  
        line 4.0  
        column 41.5  
        size 2.9 cells  
        lines 39.7 cells  
        color 229  
        font Calibri-10v00  
        id 5  
        .  
...  
*> Before displaying the screen with the toolbar, load the font to  
*> Calibri-10v00 using w$font in procedure division
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Scroll-Bar control. See "[Color management](#)" for further details.

Example - Define a scroll-bar with background and foreground color

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   background-color 14  
   foreground-color 4  
   id 5  
   .  
...
```

Help-Id

This property allows you to assign a unique ID to the Scroll-Bar control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a scroll-bar with help-id

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   font Calibri-10v00  
   help-id 3029  
   id 5  
   .  
...
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Scroll-Bar control.

Example - Define a scroll-bar with hint text

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   font Calibri-10v00  
   help-id 3029  
   id 5  
   hint "Scroll-bar hint"  
   .  
...
```

Id

This property allows you to assign a unique ID to the Scroll-Bar control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a scroll-bar with ID

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   font Calibri-10v00  
   help-id 3029  
   id 5  
   .  
...
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a scroll-bar with layout-data to resize in Y if the layout-manager requires so

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   layout-data 16  
   id 5  
   .  
...
```

Line

This property allows you to specify the Scroll-Bar control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Scroll-Bar control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Scroll-Bar control will be relative to the starting position of the prior Screen Section item.

When the Scroll-Bar control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Scroll-Bar, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a scroll-bar at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-br-1 Bar  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Scroll-Bar control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Scroll-Bar control is still computed in CELLS, but the cell size is based on the font set for the Scroll-Bar control with the [Font](#) property. If no font has been defined for the Scroll-Bar control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a scroll-bar with height defined in lines

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
    line 4.0  
    column 41.5  
    size 2.9 cells  
    lines 39.7 cells  
    color 229  
    layout-data 16  
    id 5  
    .  
...
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a scroll-bar with layout-data and maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
    line 4.0  
    column 41.5  
    size 2.9 cells  
    lines 39.7 cells  
    color 229  
    help-id 3029  
    id 5  
    max-width 2.9  
    min-width 2.9  
    min-height 60.0  
    max-height 15.0  
    layout-data 16  
    hint "Scroll-bar hint"  
    .  
...
```

Max-Val

This property is the maximum value represented by the Scroll-Bar control.

Example - Define a scroll-bar with layout-data and maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   max-width 2.9  
   min-width 2.9  
   min-height 60.0  
   max-height 15.0  
   layout-data 16  
   hint "Scroll-bar hint"  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a scroll-bar with layout-data and maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   max-width 2.9  
   min-width 2.9  
   min-height 60.0  
   max-height 15.0  
   layout-data 16  
   hint "Scroll-bar hint"  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a scroll-bar with layout-data and maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   max-width 2.9  
   min-width 2.9  
   min-height 60.0  
   max-height 15.0  
   layout-data 16  
   hint "Scroll-bar hint"  
   .  
...
```

Min-Val

This property is the minimum value represented by the Scroll-Bar control.

Example - Define a scroll-bar with maximum and minimum values

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   layout-data 16  
   min-val 2  
   max-val 20  
   .  
...
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a scroll-bar with layout-data and maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   max-width 2.9  
   min-width 2.9  
   min-height 60.0  
   max-height 15.0  
   layout-data 16  
   hint "Scroll-bar hint"  
   .  
...
```

Page-Size

This property sets the number of items in a page. This usually affects the size of the slider that represents one page of items.

Example - Define a scroll-bar with page-size

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   hint "Scroll-bar hint"  
   min-val 2  
   max-val 20  
   page-size 4  
   .  
...
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Scroll-Bar control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a scroll-bar with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   pop-up menu hmenu  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   hint "Scroll-bar hint"  
   min-val 2  
   max-val 20  
   page-size 4  
   .  
...  
procedure division.  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Scroll-Bar control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Scroll-Bar control is still computed in CELLS, but the cell size is based on the font set for the Scroll-Bar control with the [Font](#) property. If no font has been defined for the Scroll-Bar control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a scroll-bar with size property

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   hint "Scroll-bar hint"  
   min-val 2  
   max-val 20  
   page-size 4  
   .  
...
```

Value

This property represents the slider's position in the control. The range of values is set by the properties [Min-Val](#), [Max-Val](#), and [Page-Size](#).

Example - Define a scroll-bar with value property

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 4.0  
   column 41.5  
   size 2.9 cells  
   lines 39.7 cells  
   color 229  
   help-id 3029  
   id 5  
   value 10  
   min-val 2  
   max-val 20  
   page-size 4  
   .  
...
```

Visible

This property assumes a value of "0" if the Scroll-Bar control is not visible, "1" if it is visible.

Example - Modify a scroll-bar to make it invisible

```
procedure division.  
...  
   modify screen-1-sb-1 visible 0  
...
```

Styles

The following styles are applicable to the SCROLL-BAR control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Horizontal](#), [Low](#), [Lowlight](#), [Permanent](#), [Standard](#), [Temporary](#), [Track-Thumb](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a scroll-bar with low background and bold foreground

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 2.8  
   column 25.7  
   size 2.0 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Scroll-Bar control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a scroll-bar with height and width in cells

```
screen section.  
...  
03 screen-1-sb-2 Scroll-Bar  
   line 39.1  
   column 12.8  
   size 2.0  
   lines 26.8  
   id 10  
   height-in-cells  
   width-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a scroll-bar with low background and bold foreground

```
screen section.  
...  
03 screen-1-sb-1 Scroll-Bar  
   line 2.8  
   column 25.7  
   size 2.0 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   .
```

Horizontal

When this style is set, the Scroll-Bar control is horizontal.

When it is not set, the Scroll-Bar control is vertical. This is the default setting.

Example - Define a horizontal scroll-bar

```
screen section.  
...  
03 screen-1-sb-2 Scroll-Bar  
   line 39.1  
   column 12.8  
   size 38.4  
   lines 3.8  
   id 10  
   horizontal  
   .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a scroll-bar that is temporary

```
screen section.  
...  
03 screen-1-sb-2 Scroll-Bar  
   line 39.1  
   column 12.8  
   size 2.4  
   lines 30.8  
   id 10  
   temporary  
.
```

Track-Thumb

The keyword Track-Thumb is supported for compatibility but is treated as commentary. This style has no effect.

Example - Define a scroll-bar with track thumb

```
screen section.  
...  
03 screen-1-sb-2 Scroll-Bar  
   line 39.1  
   column 12.8  
   size 2.4  
   lines 30.8  
   id 10  
   track-thumb  
.
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Scroll-Bar control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a scroll-bar with height and width in cells

```
screen section.  
...  
03 screen-1-sb-2 Scroll-Bar  
   line 39.1  
   column 12.8  
   size 2.4  
   lines 30.8  
   id 10  
   height-in-cells  
   width-in-cells  
   .
```

Events

The following events are applicable to the SCROLL-BAR control: [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-SB-THUMB](#), [MSG-VALIDATE](#).

CMD-GOTO

This event is fired when the user tries to activate the Scroll-Bar control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Scroll-Bar control is requested. The EVENT-DATA-2 data item contains the Scroll-Bar control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-SB-THUMB

This event is fired when the user releases the mouse button after moving the slider in a Scroll-Bar control. The EVENT-DATA-2 data item contains the new slider position.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

SLIDER



A Slider is used to represent and retrieve a discrete value in a range. Optional tick marks can be shown to make the selection easier. Sliders can be horizontal or vertical.

Properties

The following properties are applicable to the SLIDER control: [Background-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Labels-Increment](#), [Layout-data](#), [Line](#), [Lines](#), [Major-Tick-Spacing](#), [Max-Height](#), [Max-Val](#), [Max-Width](#), [Min-Height](#), [Min-Val](#), [Min-Width](#), [Minor-Tick-Spacing](#), [Page-Size](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Size](#), [Value](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Slider control. See "[Color management](#)" for further details.

Example - Define a slider with background and foreground colors

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   background-color 6  
   foreground-color 3  
   id 9  
   horizontal  
   .
```

[Col | Column | Pos | Position]

This property allows you to specify the Slider control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Slider control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Slider control will be relative to the ending position of the

prior Screen Section item.

When the Slider control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Slider, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a slider at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-sb-1 Slider  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Slider control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a slide-bar with color property

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   color 135  
   id 9  
   horizontal  
   .
```

Css-Base-Style-Name

This property is ignored as the SLIDER control is not supported in a Web Direct 2.0 environment.

Css-Style-Name

This property is ignored as the SLIDER control is not supported in a Web Direct 2.0 environment.

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a slider

```
...  
  modify screen-1-sl-1 custom-data "Screen 1 custom data"  
...
```

Enabled

This property assumes a value of "0" if the Slider control is disabled, "1" if it is enabled.

Example - Define a slide-bar initially disabled to enable it on procedure division later

```
screen section.  
...  
  03 screen-1-sl-1 Slider  
    line 33.1  
    column 12.8  
    size 38.2 cells  
    lines 3.0 cells  
    color 135  
    id 9  
    horizontal  
    enabled 0  
    .  
...  
procedure division.  
...  
  modify screen-1-sl-1 enabled 1  
...  

```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a slide-bar with a list of events to be excluded

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   color 135  
   id 9  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   horizontal  
   .
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a slide-bar with a list of events to be excluded

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   color 135  
   id 9  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   horizontal  
   .
```

Font

This property specifies the font used to display the content of the Slider control. It may be used to compute the height and the width of the Slider control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a slide-bar with a particular font

```
working-storage section.  
77 Tahoma-10v00 handle of font.  
...  
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   font Tahoma-10v00  
   .
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Slider control. See "[Color management](#)" for further details.

Example - Define a slide-bar with foreground and background colors

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   background-color 6  
   foreground-color 3  
   id 9  
   horizontal  
   .
```

Help-Id

This property allows you to assign a unique ID to the Slider control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a slide-bar with help-id

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   help-id 7002  
   id 9  
   horizontal  
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Slider control.

Example - Define a slide-bar with a hint text

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   color 135  
   id 9  
   hint "Slide this bar to increase the height of rows in the grid below"  
   horizontal  
   .
```

Id

This property allows you to assign a unique ID to the Slider control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a slide-bar with an ID property

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 33.1  
   column 12.8  
   size 38.2 cells  
   lines 3.0 cells  
   color 135  
   id 9  
   horizontal  
   .
```

Labels-Increment

This property sets the text of the numeric labels of the Slider by creating an enumeration that starts from [Min-Val](#). For example, if you set Labels-Increment to 10 and Min-Val is zero, then the following labels will be created: 0, 10, 20, 30 ...

Example - Define a slide-bar with minimum and maximum value and the labels increment

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 32.1  
   column 12.8  
   size 38.2 cells  
   lines 4.0 cells  
   color 133  
   font Calibri-9v0  
   help-id 7002  
   id 9  
   horizontal  
   min-val 10  
   max-val 100  
   labels-increment 10  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a slide-bar with layout data to resize on X and Y when the layout manager requires and

with maximum and minimum size in width and height

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   max-width 100.0  
   min-width 40.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   horizontal  
   .
```

Line

This property allows you to specify the Slider control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Slider control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Slider control will be relative to the starting position of the prior Screen Section item.

When the Slider control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Slider, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a slider at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-sl-1 Slider  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Slider control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Slider control is still computed in CELLS, but the cell size is based on the font set for the Slider control

with the [Font](#) property. If no font has been defined for the Slider control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a slide-bar with height in lines

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   max-width 100.0  
   min-width 40.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   horizontal  
   .
```

Major-Tick-Spacing

This property sets the major increment in the range of values. When the [Show-Ticks](#) style is set, this property affects the amount of space between the major tick markers.

Example - Define a slide-bar with show ticks style and the major tick spacing

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   major-tick-spacing 3  
   labels-increment 10  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a slide-bar with layout data to resize on X and Y when the layout manager requires and

with maximum and minimum size in width and height

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   max-width 100.0  
   min-width 40.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   horizontal  
   .
```

Max-Val

This property is the maximum value represented by the Slider control.

Example - Define a slide-bar with min and max values

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   major-tick-spacing 3  
   labels-increment 10  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a slide-bar with layout data to resize on X and Y when the layout manager requires and

with maximum and minimum size in width and height

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   max-width 100.0  
   min-width 40.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   horizontal  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a slide-bar with layout data to resize on X and Y when the layout manager requires and with maximum and minimum size in width and height

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   max-width 100.0  
   min-width 40.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   horizontal  
   .
```

Min-Val

This property is the minimum value represented by the Slider control.

Example - Define a slide-bar with min and max values

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   major-tick-spacing 3  
   labels-increment 10  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a slide-bar with layout data to resize on X and Y when the layout manager requires and with maximum and minimum size in width and height

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   max-width 100.0  
   min-width 40.0  
   min-height 5.0  
   max-height 20.0  
   layout-data 17  
   horizontal  
   .
```

Minor-Tick-Spacing

This property sets the minor increment in the range of values. When the [Show-Ticks](#) style is set, this property affects the amount of space between the minor tick markers.

Example - Define a slider-bar with minor and mayor tick spacing and show ticks

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   minor-tick-spacing 1  
   major-tick-spacing 5  
   labels-increment 15  
   .
```

Page-Size

This property sets the size of the range covered by the knob.

Example - Define a slide-bar from 10 to 100 and the knob will go thru 95 only

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   minor-tick-spacing 1  
   major-tick-spacing 5  
   labels-increment 15  
   .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Slider control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a slide-bar with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-sl-1 Slider  
   pop-up menu hmenu  
   line 34.2  
   column 4.6  
   size 26.3 cells  
   lines 4.9 cells  
   id 7  
   horizontal  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Size

This property allows you to specify the size of the Slider control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Slider control is still computed in CELLS, but the cell size is based on the font set for the Slider control with the [Font](#) property. If no font has been defined for the Slider control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a slide-bar with the size property

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   .
```

Value

This property represents the value of the Slider control.

When inquired, it returns the value that is currently represented.

When set, the Slider control changes its look to represent it.

It is the position of the slider.

Example - Get the value of an slide-bar

```
working-storage section.  
77 ws-sl2-val    pic 9(5).  
...  
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   .  
...  
procedure division.  
...  
    inquire screen-1-sl-2 value ws-sl2-val  
...  

```

Visible

This property assumes a value of "0" if the Slider control is not visible, "1" if it is visible.

Example - Define a slide-bar initially invisible and make it visible on procedure division

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   visible 0  
   .  
...  
procedure division.  
...  
   modify screen-1-sl-2 visible 1  
...  

```

Styles

The following styles are applicable to the SLIDER control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Horizontal](#), [Inverted](#), [Low](#), [Lowlight](#), [Permanent](#), [Show-Labels](#), [Show-Ticks](#), [Standard](#), [Temporary](#), [Width-In-Cells](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a slide-bar with background low and foreground bold

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   background-low  
   bold  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Slider control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines value CELLS](#)".

Example - Define a slide-bar with height and width in cells

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4  
   lines 3.8  
   id 10  
   background-low  
   bold  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   height-in-cells  
   width-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a slide-bar with background low and foreground bold

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   background-low  
   bold  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   .
```

Horizontal

When this style is set, the slider bar is displayed from left to right. The [Lines](#) property controls the height and the [Size](#) property controls the width.

Without this style instead the slider bar is displayed from top to bottom. The [Lines](#) property controls the width and the [Size](#) property controls the height.

Example - Define a horizontal slide-bar

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   .
```

Inverted

When this style is set, the values are represented from top to bottom. When used in conjunction with the [Horizontal](#) style, the values are represented from right to left.

Example - Define a horizontal slide-bar where the max value appears on the left and the min value on the right (inverted)

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   inverted  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   .
```

{ Permanent | Temporary }

Permanent

A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.

Temporary

Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary slide-bar

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   temporary  
   .
```

Show-Labels

When this style is set, the value of each major tick marker is shown. To see values, the [Show-Ticks](#) style must be also set.

Example - Define a slide-bar with show-labels and show-ticks styles

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   minor-tick-spacing 1  
   major-tick-spacing 5  
   labels-increment 15  
   .
```

Show-Ticks

When this style is set, tick markers are shown.

Example - Define a slide-bar with show-labels and show-ticks styles

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4 cells  
   lines 3.8 cells  
   id 10  
   horizontal  
   show-ticks  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   minor-tick-spacing 1  
   major-tick-spacing 5  
   labels-increment 15  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Slider control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size](#) *value* CELLS".

Example - Define a slide-bar with height and width in cells

```
screen section.  
...  
03 screen-1-sl-2 Slider  
   line 39.1  
   column 12.8  
   size 38.4  
   lines 3.8  
   id 10  
   background-low  
   bold  
   horizontal  
   show-labels  
   min-val 10  
   max-val 100  
   page-size 5  
   labels-increment 15  
   height-in-cells  
   width-in-cells  
   .
```

Events

The following events are applicable to the SLIDER control: [CMD-GOTO](#), [CMD-HELP](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-SL-THUMB](#), [MSG-VALIDATE](#).

CMD-GOTO

This event is fired when the user tries to activate the Slider control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Slider control is requested. The EVENT-DATA-2 data item contains the Slider control [Help-Id](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

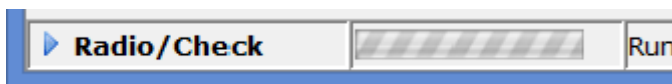
MSG-SL-THUMB

This event is fired each time the user moves the slider in a Slider control. The EVENT-DATA-2 data item contains the new slider position.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

STATUS-BAR



A Status-Bar is always shown at the bottom of the screen and it is used to show information. It can be divided

into several clickable panels, that can contain graphics, text or both.

Properties

The following properties are applicable to the STATUS-BAR control: [Background-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Layout-data](#), [Line](#), [Lines](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Panel-Alignment](#), [Panel-Background-Color](#), [Panel-Bitmap](#), [Panel-Bitmap-Alignment](#), [Panel-Bitmap-Number](#), [Panel-Bitmap-Width](#), [Panel-Color](#), [Panel-Foreground-Color](#), [Panel-Hint](#), [Panel-Index](#), [Panel-Style](#), [Panel-Text](#), [Panel-Widths](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Status-Bar control. See "[Color management](#)" for further details.

Example - Display a status-bar with 2 panels and background and foreground colors

```
working-storage section.
77 screen-1-st-2-hdl  handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Default-Font
        background-color 7
        foreground-color 3
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        handle screen-1-st-2-hdl
    .
    modify screen-1-st-2-hdl
        panel-index 1
        panel-text "Date: "
        panel-index 2
        panel-text "Screen Status: "
    .
```

[Col | Column | Pos | Position]

This property has no effect on the STATUS-BAR. The STATUS-BAR is always positioned at column 1.

Color

This property allows you to set or retrieve the color of the Status-Bar control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Display a status-bar with 2 panels and using the color property

```
working-storage section.  
77 screen-1-st-2-hdl handle of status-bar.  
...  
procedure division.  
...  
*> Display a Window first to display the status-bar upon it  
...  
    display status-bar  
        font Default-Font  
        color 257  
        panel-widths ( 30 50)  
        panel-style ( 1 2)  
        panel-alignment ( "U" "U")  
        panel-bitmap ( icon-png2 icon-png2)  
        panel-bitmap-width ( 18 18)  
        panel-bitmap-number ( 4 5)  
        panel-bitmap-alignment ( "U" "U")  
        handle screen-1-st-2-hdl  
    .  
    modify screen-1-st-2-hdl  
        panel-index 1  
        panel-text "Date: "  
        panel-index 2  
        panel-text "Screen Status: "  
    .
```

Css-Base-Style-Name

This property is ignored as the STATUS-BAR control is not supported in a Web Direct 2.0 environment.

Css-Style-Name

This property is ignored as the STATUS-BAR control is not supported in a Web Direct 2.0 environment.

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a status-bar

```
procedure division.  
...  
    modify screen-1-br-1 custom-data "Screen 1 custom data"  
...  

```

Enabled

This property is ignored by the status-bar control.

Font

This property specifies the font used to display the content of the Status-Bar control.

Example - Display a status-bar with 2 panels and special font

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
77 Calibri-10v0-b handle of font.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Calibri-10v0-b
        background-color 7
        foreground-color 3
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        handle screen-1-st-2-hdl
    .
    modify screen-1-st-2-hdl
        panel-index 1
        panel-text "Date: "
        panel-index 2
        panel-text "Screen Status: "
    .
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Status-Bar control. See "[Color management](#)" for further details.

Example - Display a status-bar with 2 panels and background and foreground colors

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-index 2
    panel-text "Screen Status: "
  .
```

Help-Id

This property allows you to assign a unique ID to the Status-Bar control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Display a status-bar with 2 panels and help id

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Default-Font
        background-color 7
        foreground-color 3
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        help-id 4
        handle screen-1-st-2-hdl
    .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a status-bar with layout-data that allows X and Y resize if the layout manager requires so

and with maximum and minimum heights

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    lines 3.5
    layout-data 17
    max-height 4.5
    min-height 1.5
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-index 2
    panel-text "Screen Status: "
  .
```

Line

This property has no effect on the STATUS-BAR. The STATUS-BAR is always positioned at the bottom of the screen.

Lines

This property allows you to specify the Status-Bar control's height. The value is specified in cells. Decimal values are allowed.

Example - Define a taller status-bar using the lines property

```
working-storage section.  
77 screen-1-st-2-hdl handle of status-bar.  
...  
procedure division.  
...  
*> Display a Window first to display the status-bar upon it  
...  
    display status-bar  
        lines 3.5  
        font Default-Font  
        background-color 7  
        foreground-color 3  
        panel-widths ( 30 50)  
        panel-style ( 1 2)  
        panel-alignment ( "U" "U")  
        panel-bitmap ( icon-png2 icon-png2)  
        panel-bitmap-width ( 18 18)  
        panel-bitmap-number ( 4 5)  
        panel-bitmap-alignment ( "U" "U")  
        handle screen-1-st-2-hdl  
    .  
    modify screen-1-st-2-hdl  
        panel-index 1  
        panel-text "Date: "  
        panel-index 2  
        panel-text "Screen Status: "  
    .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a status-bar with layout-data that allows X and Y resize if the layout manager requires so

and with maximum and minimum heights

```
working-storage section.  
77 screen-1-st-2-hdl handle of status-bar.  
...  
procedure division.  
...  
*> Display a Window first to display the status-bar upon it  
...  
    display status-bar  
        lines 3.5  
        layout-data 17  
        max-height 4.5  
        min-height 1.5  
        font Default-Font  
        background-color 7  
        foreground-color 3  
        panel-widths ( 30 50)  
        panel-style ( 1 2)  
        panel-alignment ( "U" "U")  
        panel-bitmap ( icon-png2 icon-png2)  
        panel-bitmap-width ( 18 18)  
        panel-bitmap-number ( 4 5)  
        panel-bitmap-alignment ( "U" "U")  
        handle screen-1-st-2-hdl  
    .  
    modify screen-1-st-2-hdl  
        panel-index 1  
        panel-text "Date: "  
        panel-index 2  
        panel-text "Screen Status: "  
    .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a status-bar with layout-data that allows X and Y resize if the layout manager requires so

and with maximum and minimum heights

```
working-storage section.  
77 screen-1-st-2-hdl handle of status-bar.  
...  
procedure division.  
...  
*> Display a Window first to display the status-bar upon it  
...  
    display status-bar  
        lines 3.5  
        layout-data 17  
        max-height 4.5  
        min-height 1.5  
        font Default-Font  
        background-color 7  
        foreground-color 3  
        panel-widths ( 30 50)  
        panel-style ( 1 2)  
        panel-alignment ( "U" "U")  
        panel-bitmap ( icon-png2 icon-png2)  
        panel-bitmap-width ( 18 18)  
        panel-bitmap-number ( 4 5)  
        panel-bitmap-alignment ( "U" "U")  
        handle screen-1-st-2-hdl  
    .  
    modify screen-1-st-2-hdl  
        panel-index 1  
        panel-text "Date: "  
        panel-index 2  
        panel-text "Screen Status: "  
    .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a status-bar with layout-data that allows X and Y resize if the layout manager requires so

and with maximum and minimum heights

```
working-storage section.
77 screen-1-st-2-hdl  handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    lines 3.5
    layout-data 17
    max-height 4.5
    min-height 1.5
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-index 2
    panel-text "Screen Status: "
  .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a status-bar with layout-data that allows X and Y resize if the layout manager requires so

and with maximum and minimum heights

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    lines 3.5
    layout-data 17
    max-height 4.5
    min-height 1.5
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-index 2
    panel-text "Screen Status: "
  .
```

Panel-Alignment

This property is used to set the appearance of the panel identified by the [Panel-Index](#) property. The following values can be assigned.

"C"	The text in the panel is centered.
"L"	The text in the panel is left justified.
"R"	The text in the panel is right justified.

By default, panel text is left justified.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Define a status-bar with 2 panels, defining the alignment of each one

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-index 2
    panel-text "Screen Status: "
  .
```

Panel-Background-Color

This property allows you to set or retrieve the background color of the panel identified by the [Panel-Index](#) property. See "[Color management](#)" for further details.

Example - Display a status-bar with 2 panels and include foreground and background colors for 1 panel

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-foreground-color 15
    panel-background-color 6
    panel-index 2
    panel-text "Screen Status: "
  .
```

Panel-Bitmap

Assigning a bitmap handle to this property causes an image to be shown in the panel identified by the [Panel-Index](#) property.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Display a status-bar with 2 panels having each one a bitmap

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
77 icon-png2 pic s9(9) comp-4.
...
procedure division.
...
*> Load the bitmaps prior to use them on the status-bar, use W$BITMAP routine
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Calibri-10v0-b
        color 257
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        handle screen-1-st-2-hdl
    .
    modify screen-1-st-2-hdl
        panel-index 1
        panel-text "Date: "
        panel-index 2
        panel-text "Screen Status: "
    .
```

Panel-Bitmap-Alignment

This property is used to set the placement of the panel bitmap, if any, in relation to the panel text. Valid values are:

"C"	The bitmap is centered. Bitmap and text are always overlapped and the text is always above the bitmap.
"L"	The text is placed to the left of the bitmap. Bitmap and text do not overlap.
"R"	The text is placed to the right of the bitmap. Bitmap and text do not overlap.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Display a status-bar with 2 panels, having a bitmap aligned to the right each one

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
77 icon-png2 pic s9(9) comp-4.
...
procedure division.
...
*> Load the bitmaps prior to use them on the status-bar, use W$BITMAP routine
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Calibri-10v0-b
    color 257
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "R" "R")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-index 2
    panel-text "Screen Status: "
  .
```

Panel-Bitmap-Number

This property defines which bitmap among the ones in the bitmap strip referenced by the [Panel-Bitmap](#) property is to be displayed in the panel identified by the [Panel-Index](#) property.

When the values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Display a status-bar with 2 panels having each one a different bitmap image from the same bitmap

file

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
77 icon-png2 pic s9(9) comp-4.
...
procedure division.
...
*> Load the bitmaps prior to use them on the status-bar, use W$BITMAP routine
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Calibri-10v0-b
        color 257
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        handle screen-1-st-2-hdl
    .
    modify screen-1-st-2-hdl
        panel-index 1
        panel-text "Date: "
        panel-index 2
        panel-text "Screen Status: "
    .
```

Panel-Bitmap-Width

This property defines the width in pixels of the image displayed in the panel identified by the [Panel-Index](#) property.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Display a status-bar with 2 panels having each one a different bitmap image from the same bitmap

file, specifying the bitmap width within the file

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
77 icon-png2 pic s9(9) comp-4.
...
procedure division.
...
*> Load the bitmaps prior to use them on the status-bar, use W$BITMAP routine
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Calibri-10v0-b
        color 257
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        handle screen-1-st-2-hdl
    .
    modify screen-1-st-2-hdl
        panel-index 1
        panel-text "Date: "
        panel-index 2
        panel-text "Screen Status: "
    .
```

Panel-Color

This property allows you to set or retrieve the color of the panel identified by the [Panel-Index](#) property. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Display a status-bar with 2 panels and include foreground and background colors for 1 panel in

one single color property

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
  .
```

Panel-Foreground-Color

This property allows you to set or retrieve the foreground color of the panel identified by the [Panel-Index](#) property. See "[Color management](#)" for further details.

Example - Display a status-bar with 2 panels and include foreground and background colors for 1 panel

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-foreground-color 15
    panel-background-color 6
    panel-index 2
    panel-text "Screen Status: "
  .
```

Panel-Hint

This property is used to set the tooltip shown in the panel identified by the [Panel-Index](#) property.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Display a status-bar with 2 panels and hint text for each panel

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

Panel-Index

This property represents the panel that the following properties refer to: [Panel-Alignment](#), [Panel-Bitmap](#), [Panel-Bitmap-Number](#), [Panel-Bitmap-Width](#), [Panel-Style](#), [Panel-Text](#), and [Panel-Widths](#).

Example - Display a status-bar with 2 panels each one with its index

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

Panel-Style

This property is used to set the appearance of the panel identified by the [Panel-Index](#) property. The following symbolic values, included in the copy file [isgui.def](#), can be assigned.

panel-flat	The panel has no border and appears at the same level of the window.
panel-lowered	The panel has a border that makes it appear to be lowered.
panel-raised	The panel has a border that makes it appear to be raised.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Define a status-bar with 2 panels, the first lowered and the second raised

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

Panel-Text

This property is used to set the text shown in the panel identified by the [Panel-Index](#) property.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Example - Define a status-bar with 2 panels, each one with different text

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

Panel-Widths

This property is used to set the width, in cells, of the panel identified by the [Panel-Index](#) property.

When values are enclosed between parentheses, the [Panel-Index](#) property is ignored, the 1st value refers to the 1st panel, the 2nd value refers to the 2nd panel and so on.

Setting this property to 0 creates a status bar with one panel extending across its entire width and no text and reset existing panels, if any.

When the status-bar is displayed on a resizable window, when the window shrinks below the size specified by Panel-Widths, all the panels are reset to an average size small enough so that all panels fit in the window. If you do not want the all of the panels truncated, set the width of the last panel to "-1", this will cause the last panel to be sized to whatever space is left available on the window after the first panels have been accommodated.

Example - Define a status-bar with 2 panels, the former has fixed width while the latter extends to the end of

the window

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 -1)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Status-Bar control by assigning a pop-up menu handle to it. The [MSG-ST-DBLCLICK](#) event may be generated.

Example - Define a status-bar with 2 panels and a pop-up menu

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
77 hmenu                pic s9(9) comp.4.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        font Default-Font
        background-color 7
        foreground-color 3
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        pop-up menu hmenu
        handle screen-1-st-2-hdl
    .
*> Use w$menu to build the pop-up menu
```

Size

This property has no effect on the STATUS-BAR. The STATUS-BAR size always matches with the size of the container window.

Visible

This property assumes a value of "0" if the Status-Bar control is not visible, "1" if it is visible.

Example - Display a status-bar, initially invisible to make it visible later on

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
01 filler pic x.
   88 show-status-bar value "y" false "n".

...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
   display status-bar
       visible 0
       font Default-Font
       background-color 7
       foreground-color 3
       panel-widths ( 30 50)
       panel-style ( 1 2)
       panel-alignment ( "U" "U")
       panel-bitmap ( icon-png2 icon-png2)
       panel-bitmap-width ( 18 18)
       panel-bitmap-number ( 4 5)
       panel-bitmap-alignment ( "U" "U")
       handle screen-1-st-2-hdl
       .
   modify screen-1-st-2-hdl
       panel-index 1
       panel-text "Date: "
       panel-hint "This is the current system date"
       panel-color 516
       panel-index 2
       panel-text "Screen Status: "
       panel-hint "This is the status of the program"
       .
...
   if show-status-bar
       modify screen-1-st-2-hdl visible 1
   end-if
...

```

Styles

The following styles are applicable to the STATUS-BAR control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Grip](#), [High](#), [Highlight](#), [Low](#), [Lowlight](#), [Permanent](#), [Standard](#), [Temporary](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Display a status-bar with low background and bold foreground

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
    display status-bar
        background-low
        bold
        font Default-Font
        background-color 7
        foreground-color 3
        panel-widths ( 30 50)
        panel-style ( 1 2)
        panel-alignment ( "U" "U")
        panel-bitmap ( icon-png2 icon-png2)
        panel-bitmap-width ( 18 18)
        panel-bitmap-number ( 4 5)
        panel-bitmap-alignment ( "U" "U")
        handle screen-1-st-2-hdl
    .
    modify screen-1-st-2-hdl
        panel-index 1
        panel-text "Date: "
        panel-hint "This is the current system date"
        panel-color 516
        panel-index 2
        panel-text "Screen Status: "
        panel-hint "This is the status of the program"
    .
```

Grip

When this style is set, a triangle is displayed in the lower right corner of the Status-Bar control. This is usually done to notify the user that the window is resizable and that he can resize it by dragging the rightmost part of the Status-Bar control.

Note: the triangle is shown only if the Window has the [Resizable](#) style.

Example - Display a status-bar with grip style

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    grip
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Display a status-bar with low background and bold foreground

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    background-low
    bold
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Display a status-bar that is temporary

```
working-storage section.
77 screen-1-st-2-hdl handle of status-bar.
...
procedure division.
...
*> Display a Window first to display the status-bar upon it
...
  display status-bar
    background-low
    temporary
    font Default-Font
    background-color 7
    foreground-color 3
    panel-widths ( 30 50)
    panel-style ( 1 2)
    panel-alignment ( "U" "U")
    panel-bitmap ( icon-png2 icon-png2)
    panel-bitmap-width ( 18 18)
    panel-bitmap-number ( 4 5)
    panel-bitmap-alignment ( "U" "U")
    handle screen-1-st-2-hdl
  .
  modify screen-1-st-2-hdl
    panel-index 1
    panel-text "Date: "
    panel-hint "This is the current system date"
    panel-color 516
    panel-index 2
    panel-text "Screen Status: "
    panel-hint "This is the status of the program"
  .
```

Events

The following events are applicable to the STATUS-BAR control: [CMD-HELP](#), [MSG-ST-DBLCLICK](#).

CMD-HELP

This event is fired when the help for the Status-Bar control is requested. The EVENT-DATA-2 data item contains the Status-Bar control [Help-Id](#).

MSG-ST-DBLCLICK

This event is fired when the user double-clicks a panel of the Status-Bar control. EVENT-DATA-1 contains the panel that the user double-clicked.

TAB-CONTROL



A Tab-Control holds one or more Tabs that can be activated by the user, one at a time. It is commonly used to display several pages of controls on the same screen. When a Tab is clicked, the corresponding page is made visible. Tabs can contain graphics, text or both.

Properties

The following properties are applicable to the TAB-CONTROL control: [Background-Color](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [Font](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Insertion-Index](#), [Line](#), [Layout-data](#), [Lines](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Reset-Tabs](#), [Size](#), [Tab-Alignment](#), [Tab-Enabled](#), [Tab-Index](#), [Tab-Text](#), [Tab-To-Add](#), [Tab-To-Delete](#), [Value](#), [Visible](#).

Background-Color

This property allows you to set or retrieve the background color of the Tab-Control control. See "[Color management](#)" for further details.

If the [Allow-Container](#) style is set, then both tab area and tab labels are colored, otherwise only the tab labels are colored.

Example - Define a tab control with background and foreground color

```
...
screen section.
...
03 screen-1-tc-1 Tab-Control
   line 2.8
   column 25.7
   size 34.6 cells
   lines 28.9 cells
   background-color 7
   foreground-color 3
   id 10
   bitmap-width 16
   .
...
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used in the Tab identified by the [Tab-Index](#) property.

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Define a tab-control with a bitmap

```
working-storage section.
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-tc-1 Tab-Control
   line 2.8
   column 25.7
   size 34.6 cells
   lines 28.9 cells
   background-color 7
   foreground-color 3
   id 10
   bitmap-width 18
   bitmap-handle icon-png
   .
03 screen-1-tp-1 visible screen-1-tp-1-vis.
05 screen-1-ef-1 Entry-Field
   line 10.7
   column 35.6
   size 12.3 cells
   lines 5.3 cells
   id 11
   3-d
   .
05 screen-1-pb-9 Push-Button
   line 17.7
   column 35.4
   size 12.7 cells
   lines 4.4 cells
   id 12
   title "Push-Button"
   .
03 screen-1-tp-2 visible screen-1-tp-2-vis.
05 screen-1-ef-2 Entry-Field
   line 11.8
   column 35.1
   size 15.3 cells
   lines 4.8 cells
   id 13
   3-d
   .
05 screen-1-pb-10 Push-Button
   line 19.5
   column 35.0
   size 15.6 cells
   lines 5.0 cells
   id 14
   title "Push-Button"
   .
```

Bitmap-Number

This property defines which image among the ones in the bitmap strip referenced by the [Bitmap-Handle](#) property is to be displayed in the page identified by the [Tab-Index](#) property.

When the values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Modify a tab-control to assign a bitmap number to each page

```
procedure division.  
...  
    modify screen-1-tc-1  
        tab-to-add ( "Page-1" "Page-2")  
        bitmap-number ( 3 2)  
    .  
...
```

Bitmap-Width

This property defines the width in pixels of the image displayed on the Tab identified by the [Tab-Index](#) property. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Define a tab-control with a bitmap and its bitmap width

```
working-storage section.
77 icon-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-tc-1 Tab-Control
   line 2.8
   column 25.7
   size 34.6 cells
   lines 28.9 cells
   background-color 7
   foreground-color 3
   id 10
   bitmap-width 18
   bitmap-handle icon-png
.
03 screen-1-tp-1 visible screen-1-tp-1-vis.
05 screen-1-ef-1 Entry-Field
   line 10.7
   column 35.6
   size 12.3 cells
   lines 5.3 cells
   id 11
   3-d
.
05 screen-1-pb-9 Push-Button
   line 17.7
   column 35.4
   size 12.7 cells
   lines 4.4 cells
   id 12
   title "Push-Button"
.
03 screen-1-tp-2 visible screen-1-tp-2-vis.
05 screen-1-ef-2 Entry-Field
   line 11.8
   column 35.1
   size 15.3 cells
   lines 4.8 cells
   id 13
   3-d
.
05 screen-1-pb-10 Push-Button
   line 19.5
   column 35.0
   size 15.6 cells
   lines 5.0 cells
   id 14
   title "Push-Button"
.
```

[Col | Column | Pos | Position]

This property allows you to specify the Tab-Control control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Tab-Control control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Tab-Control control will be relative to the ending position of the prior Screen Section item.

When the Tab-Control control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03  Label, COL 2, SIZE 12, (more screen options).  
03  Tab-Control, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a tab-control at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Color

This property allows you to set or retrieve the color of the Tab-Control control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

If the [Allow-Container](#) style is set, then both tab area and tab labels are colored, otherwise only the tab labels are colored.

Example - Define a tab-control with foreground and background colors

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   color 136  
   id 10  
   bitmap-width 18  
   bitmap-handle icon-png  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a tab-control with `css-base-style-name`, applicable with WD2

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   css-base-style-name "css-tabcontrol"  
   bitmap-width 18  
   bitmap-handle icon-png  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a tab-control with `css-style-name`, applicable with WD2

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   css-style-name "css-tabcontrol"  
   bitmap-width 18  
   bitmap-handle icon-png  
   .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a tab-control

```
procedure division.  
...  
   modify screen-1-tc-1 custom-data "Screen 1 custom data"  
...  
...
```

Enabled

This property assumes a value of "0" if the Tab-Control control is disabled, "1" if it is enabled.

Example - Define a tab-control initially disabled

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   enabled 0  
   id 10  
   bitmap-width 18  
   bitmap-handle icon-png  
   .
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a tab-control with an excluded event-list

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   enabled 0  
   id 10  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   .
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a tab-control with an excluded event-list

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   enabled 0  
   id 10  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   .
```

Font

This property specifies the font used to display the content of the Tab-Control control. It may be used to compute the height and the width of the Tab-Control control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a tab-control with a particular Font

```
working-storage section.  
77 Tahoma-10v00 handle of font.  
...  
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   font Tahoma-10v00  
   enabled 0  
   id 10  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   bitmap-width 18  
   bitmap-handle icon-png  
   .
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Tab-Control control. See "[Color management](#)" for further details.

If the [Allow-Container](#) style is set, then both tab area and tab labels are colored, otherwise only the tab labels are colored.

Example - Define a tab control with background and foreground color

```
...
screen section.
...
03 screen-1-tc-1 Tab-Control
   line 2.8
   column 25.7
   size 34.6 cells
   lines 28.9 cells
   background-color 7
   foreground-color 3
   id 10
   bitmap-width 16
   .
...
```

Help-Id

This property allows you to assign a unique ID to the Tab-Control control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a tab-control with help-id number

```
screen section.
...
03 screen-1-tc-1 Tab-Control
   line 2.8
   column 25.7
   size 34.6 cells
   lines 28.9 cells
   enabled 0
   help-id 1036
   id 10
   .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Tab-Control control.

Example - Define a tab-control with hint text

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   hint "Tab Control Hint"  
   .
```

Id

This property allows you to assign a unique ID to the Tab-Control control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a tab-control with ID

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   .
```

Insertion-Index

This numeric property affects the position where a new Tab is added to a Tab-Control control when the [Tab-To-Add](#) property is set. If this property is set to a positive value, the Tab is inserted immediately before the Tab the property refers to. When set to 0, the new Tab is appended after the last existing Tab.

Example - Add one page to a tab-control

```
...  
procedure division.  
...  
   modify screen-1-tc-1 insertion-index 2  
       tab-to-add "Other Page"  
...  

```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values,

depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a tab-control with layout-data to resize in X and Y when the layout manager requires so

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   layout-data 17  
   .
```

Line

This property allows you to specify the Tab-Control control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Tab-Control control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Tab-Control control will be relative to the starting position of the prior Screen Section item.

When the Tab-Control control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Tab-Control, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a tab-control at line 8.0 on the screen section definition

```
03 screen-1-tc-1 Tab-Control  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Tab-Control control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Tab-Control control is still computed in CELLS, but the cell size is based on the font set for the Tab-Control control with the [Font](#) property. If no font has been defined for the Tab-Control control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a tab-control with height in lines

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   layout-data 17  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a tab-control with maximum and minimum dimensions to be used when the layout manager requires so

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   max-width 70.0  
   min-width 17.0  
   min-height 15.0  
   max-height 60.0  
   layout-data 17  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a tab-control with maximum and minimum dimensions to be used when the layout

manager requires so

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   max-width 70.0  
   min-width 17.0  
   min-height 15.0  
   max-height 60.0  
   layout-data 17  
   .
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a tab-control with maximum and minimum dimensions to be used when the layout manager requires so

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   max-width 70.0  
   min-width 17.0  
   min-height 15.0  
   max-height 60.0  
   layout-data 17  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a tab-control with maximum and minimum dimensions to be used when the layout

manager requires so

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   max-width 70.0  
   min-width 17.0  
   min-height 15.0  
   max-height 60.0  
   layout-data 17  
   .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Tab-Control control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a tab-control with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   pop-up menu hmenu  
   line 26.8  
   column 32.4  
   size 20.4 cells  
   lines 12.8 cells  
   id 8  
   bitmap-width 16  
   .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Reset-Tabs

When set to a value greater than zero, all Tabs are removed.

Example - Modify a tab-control to remove all pages

```
procedure division.  
...  
   modify screen-1-tc-1 reset-tabs 1  
...  

```

Size

This property allows you to specify the size of the Tab-Control control. If the PIXEL keyword follows the value specified here, the size is computed in pixels. If either the CELLS keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Tab-Control control is still computed in CELLS, but the cell size is based on the font set for the Tab-Control control with the [Font](#) property. If no font has been defined for the Tab-Control control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a tab-control with size

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
.
```

Tab-Alignment

This property defines the title text alignment of the page identified by the [Tab-Index](#) property in a tab-control with the [Accordion](#) style.

Possible values are:

C	The text is centered (default)
L	The text is left aligned
R	The text is right aligned

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

```
TAB-ALIGNMENT = ( "L", "C" )
```

When set to space or spaces, the list is reset.

When a single value other than space is set, it is appended to the list. This is useful to define a user-defined appearance.

Example - Make the title of the second page right aligned

```
procedure division.  
...  
    modify screen-1-tc-1 tab-index 2 tab-alignment "R"  
...  

```

Tab-Enabled

This property specifies whether the page identified by the TAB-INDEX property can be selected by the user or not. A value of 0 means that the page is disabled, so clicking on that page will produce no effects. A value of 1 means that the page is enabled, so the user can select it. Each page is enabled by default.

Example - Modify a tab-control to disable page 2

```
procedure division.  
...  
    modify screen-1-tc-1 tab-index 2 tab-enabled 0  
...  

```

Tab-Index

This property represents the Tab that the following properties refer to: [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), and [Tab-Text](#).

Example - Modify a tab-control to disable page 2

```
procedure division.  
...  
    modify screen-1-tc-1 tab-index 2 tab-enabled 0  
...  

```

Tab-Text

This property defines the title of the page identified by the [Tab-Index](#) property.

When values are enclosed between parentheses, the [Tab-Index](#) property is ignored, the 1st value refers to the 1st Tab, the 2nd value refers to the 2nd Tab and so on.

Example - Get the text of the second page of a tab-control

```
procedure division.  
...  
    inquire screen-1-tc-1 tab-index 2 tab-text w-page-title  
...  

```

Tab-To-Add

When this property is set, a new Tab is added to the Tab-Control control. The value of this property represents the title of the new Tab.

Multiple Tabs can be added at the same time, by enclosing their titles between parentheses.

Example - Add one page to a tab-control

```
...  
procedure division.  
...  
    modify screen-1-tc-1 insertion-index 2  
        tab-to-add "Other Page"  
...  

```

Tab-To-Delete

When set to a positive value, the corresponding Tab is removed from the Tab-Control control.

Example - Remove the first page of a tab-control

```
procedure division.  
...  
    modify screen-1-tc-1 tab-to-delete 1  
...  

```

Value

This property represents the value of the Tab-Control control.

When inquired, it returns the value that is currently represented.

When set, the Tab-Control control changes its look to represent it.

It is the currently selected Tab.

Example - Get the selected page number

```
...  
procedure division.  
...  
    inquire screen-1-tc-1 value ws-tc-1  
...  

```

Visible

This property assumes a value of "0" if the Tab-Control control is not visible, "1" if it is visible. The visibility is applied to the whole control, it's not possible to hide and show the individual pages.

Example - Modify a tab-control to make it invisible

```
...  
procedure division.  
...  
    modify screen-1-tc-1 visible 0  
...
```

Styles

The following styles are applicable to the TAB-CONTROL control: [Accordion](#), [Allow-Container](#), [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Bottom](#), [Buttons](#), [Flat-Buttons](#), [Fixed-Width](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Hot-Track](#), [Low](#), [Lowlight](#), [Multiline](#), [No-Dividers](#), [Permanent](#), [Relative-Offset](#), [Standard](#), [Temporary](#), [Text-Norotate](#), [Vertical](#), [Width-In-Cells](#).

Accordion

When this style is set, the Tab-Control is shown as an Accordion container.

The Accordion layout is always the same, with buttons whose title is horizontal that scroll up and down when they're clicked. [Bottom](#) and [Vertical](#) styles are ignored.

This style implicitly adds the [Allow-Container](#) style.

Example - Define an accordion with two pages

```
screen section.  
...  
03 screen-1-tc-1  
   tab-control  
   line           2  
   col           2  
   lines         17 cells  
   size          68 cells  
   accordion  
   .  
03 screen-1-tc-1-page1  
   tab-group Tbl-container tab-group-value 1.  
05 label  
   line         4  
   col          4  
   title        "This is the first page"  
   .  
05 entry-field  
   line         6  
   col          4  
   .  
03 screen-1-tc-1-page2  
   tab-group Tbl-container tab-group-value 2.  
05 label  
   line         4  
   col          4  
   title        "This is the second page"  
   .  
05 combo-box  
   line         6  
   col          4  
   .
```

Allow-Container

When this style is set, the Tab-Control becomes a real container control. There is no need to manage the tab switch by intercepting the CMD-TABCHANGED event and updating the screen with the DISPLAY verb. The tab switch is automatically managed by the runtime Framework. It simplifies the coding and also reduces traffic in thin client environment.

There are two ways to add controls to a Tab-Control with the Allow-Container style:

- by using a Format 2 DISPLAY statement. E.g.

```
display scr-page-1 upon screen1-tc-1(1)
```

or

- by setting the attributes TAB-GROUP and TAB-GROUP-VALUE on the screen entry. E.g.

```
03 screen1-tc-1 tab-control
   tab-to-add ("Page1", "Page2")
   allow-container.
03 scr-page-1 tab-group screen1-tc-1, tab-group-value 1.
05 entry-field
   ...
03 scr-page-2 tab-group screen1-tc-1, tab-group-value 2.
05 list-box
   ...
```

By default, LINE and COLUMN coordinates of controls displayed on the Tab-Control are relative to the window top-left border. To make them relative to the tab top-left border, apply the [Relative-Offset](#) style as well.

Example - Define a tab-control with two pages with the allow-container style

```
screen section.
...
03 screen-1-tc-1
   tab-control
   line           2
   col            2
   lines          17 cells
   size           68 cells
   allow-container
   .
03 screen-1-tc-1-page1
   tab-group Tb1-container tab-group-value 1.
05 label
   line           4
   col            4
   title          "This is the first page"
   .
05 entry-field
   line           6
   col            4
   .
03 screen-1-tc-1-page2
   tab-group Tb1-container tab-group-value 2.
05 label
   line           4
   col            4
   title          "This is the second page"
   .
05 combo-box
   line           6
   col            4
   .
```


{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a tab-control with low background and bold foreground

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   .
```

Bottom

When this style is set, the Tabs are positioned at the bottom. When the [Vertical](#) style is also set, the Tabs are positioned on the right side.

Example - Define a tab-control with bottom style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   id 10  
   bottom  
   .
```

Buttons

This style has no effect, it's only supported for compatibility with other COBOLs.

Flat-Buttons

This style has no effect, it's only supported for compatibility with other COBOLs.

Fixed-Width

This style has no effect, it's only supported for compatibility with other COBOLs.

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Tab-Control control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines](#) value CELLS".

Example - Define a tab-control with height in cells

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9  
   id 10  
   height-in-cells  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a tab-control with low background and bold foreground

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   .
```

Hot-Track

This style has no effect, it's only supported for compatibility with other COBOLs.

Multiline

When this style is set, the Tabs are displayed on multiple lines.

Example - Define a tab-control with multiline style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   multiline  
   .
```

No-Dividers

This style has no effect, it's only supported for compatibility with other COBOLs.

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a tab-control with temporary style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   temporary  
   .
```

Relative-Offset

When this style is set, the LINE and COLUMN coordinates of controls displayed on a Tab-Control with the [Allow-Container](#) style are relative to the Tab-Control area. If the style is not set, then coordinates are relative to the window area.

Example - Define a tab-control with relative-offset style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   background-low  
   bold  
   allow-container  
   relative-offset  
   .
```

Text-Norotate

When this style is set along with the [Vertical](#) style, the text in the tab is shown horizontally.

Example - Define a tab-control with vertical and text-norotate style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   text-norotate  
   vertical  
   .
```

Vertical

When this style is set, the Tabs are positioned on the left side. When the [Bottom](#) style is also set, the Tabs are positioned on the right side.

Example - Define a tab-control with vertical and text-norotate style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6 cells  
   lines 28.9 cells  
   help-id 1036  
   id 10  
   text-norotate  
   vertical  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Tab-Control control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a tab-control with width-in-cells style

```
screen section.  
...  
03 screen-1-tc-1 Tab-Control  
   line 2.8  
   column 25.7  
   size 34.6  
   lines 28.9  
   help-id 1036  
   id 10  
   width-in-cells  
   height-in-cells  
   .
```

Events

The following events are applicable to the TAB-CONTROL control: [CMD-HELP](#), [CMD-TABCHANGED](#), [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-VALIDATE](#).

CMD-HELP

This event is fired when the help for the Tab-Control control is requested. The EVENT-DATA-2 data item contains the Tab-Control control [Help-Id](#).

CMD-TABCHANGED

This event is fired when the user selects a different Tab in the Tab-Control control. The EVENT-DATA-1 data item contains the index of the Tab that has been clicked. Before activating the new Tab, the program executes the After Procedure or the Exception Procedure associated with the current control.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item,

after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

TOOL-BAR



A Tool-Bar is a container of controls. It appears at the top of the window and usually shows Push-Buttons that the user can click. Since controls in Tool-Bars cannot be activated with the keyboard, the Tool-Bar should contain only shortcuts to commands that can be activated elsewhere with the keyboard. Usually, it replicates menu functions.

Properties

The following properties are applicable to the TOOL-BAR control: [Background-Color](#), [Cell Height](#), [Cell Size](#), [Cell Width](#), [Color](#), [Control Font](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Foreground-Color](#), [Help-Id](#), [Hint](#), [Id](#), [Layout-Manager](#), [Lines](#), [Pop-Up Menu](#).

Background-Color

This property allows you to set or retrieve the background color of the Tool-Bar control.

The value set here is also the default background color of the controls created in the Tool-Bar control.

See "[Color management](#)" for further details.

Example - Define a tool-bar with 3 buttons and foreground and background color

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
        .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
        .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
        .  
...  
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        background-color 6  
        foreground-color 7  
        lines 4.0  
        handle screen-1-tb-1-hdl  
        .  
    display screen-1-tb-1 upon screen-1-tb-1-hdl.  
...
```

Cell Height

This property defines the height of the cell used for positioning the various controls inside the Tool-Bar control.

The value must comply with the following specification:

```
{ Pixels }  
{ ControlType FONT [FontHandle] [SEPARATE] }  
{ [OVERLAPPED] }
```

Refer to the [Cell Size](#) property for a more detailed explanation.

Example - Define a tool-bar with 3 buttons and cell width and height

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
    .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
    .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
    .  
...  
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        background-color 6  
        foreground-color 7  
        lines 4.0  
        handle screen-1-tb-1-hdl  
    .  
    display screen-1-tb-1 upon screen-1-tb-1-hdl.  
...  

```

Cell Size

This property defines the size (both height and width) of the cell used for positioning the various controls inside the Tool-Bar control.

The value must comply with the following specification:

```
{ ControlType FONT [FontHandle] [SEPARATE] }  
{ [OVERLAPPED] }
```

The cell size is the space the ControlType needs to show the character "0" without truncating it. ControlType

can be LABEL, the default, or ENTRY-FIELD. When ENTRY-FIELD is specified, the cell is larger.

FontHandle is the font used to calculate the cell size. If not specified, the font set in the [Control Font](#) property will be used.

When the SEPARATE clause is specified, the cell size is increased, so that controls placed on contiguous lines or columns are not too close to each other.

When the OVERLAPPED clause is specified, the cell size is left unchanged and controls placed on contiguous lines or columns look joint. This is the default setting.

Example - Define a tool-bar with 3 buttons and cell size

```
screen section.
...
01 screen-1-tb-1.
    03 screen-1-pb-4 Push-Button
        line 1.7
        column 2.4
        size 5.2 cells
        lines 2.6 cells
        id 11
        title "Exit"
        .
    03 screen-1-pb-5 Push-Button
        line 1.7
        column 13.0
        size 5.2 cells
        lines 2.6 cells
        id 12
        title ">"
        .
    03 screen-1-pb-6 Push-Button
        line 1.7
        column 20.1
        size 5.2 cells
        lines 2.6 cells
        id 13
        title "<"
        .
...
procedure division.
...
    display tool-bar
        cell size 12
        control font Default-Font
        background-color 6
        foreground-color 7
        lines 4.0
        handle screen-1-tb-1-hdl
        .
    display screen-1-tb-1 upon screen-1-tb-1-hdl.
...

```

Cell Width

This property defines the width of the cell used for positioning the various controls inside the Tool-Bar

control.

The value must comply with the following specification:

```
{ Pixels }  
{ ControlType FONT [FontHandle] [SEPARATE] }  
{ [OVERLAPPED] }
```

Refer to the [Cell Size](#) property for a more detailed explanation.

Example - Define a tool-bar with 3 buttons and cell width and height

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
    .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
    .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
    .  
...  
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        background-color 6  
        foreground-color 7  
        lines 4.0  
        handle screen-1-tb-1-hdl  
    .  
    display screen-1-tb-1 upon screen-1-tb-1-hdl.  
...  

```

Color

This property allows you to set or retrieve the color of the Tool-Bar control.

The value set here is also the default color of the controls created in the Tool-Bar control.

Foreground and background color values are combined and therefore RGB colors are not supported. See ["Color management"](#) for further details.

Example - Define a tool-bar with 3 buttons and color property

```
screen section.  
...  
01 screen-1-tb-1.  
03 screen-1-pb-4 Push-Button  
   line 1.7  
   column 2.4  
   size 5.2 cells  
   lines 2.6 cells  
   id 11  
   title "Exit"  
   .  
03 screen-1-pb-5 Push-Button  
   line 1.7  
   column 13.0  
   size 5.2 cells  
   lines 2.6 cells  
   id 12  
   title ">"  
   .  
03 screen-1-pb-6 Push-Button  
   line 1.7  
   column 20.1  
   size 5.2 cells  
   lines 2.6 cells  
   id 13  
   title "<"  
   .  
...  
procedure division.  
...  
   display tool-bar  
       cell width 10  
       cell height 10  
       control font Default-Font  
       color 232  
       lines 4.0  
       handle screen-1-tb-1-hdl  
   .  
   display screen-1-tb-1 upon screen-1-tb-1-hdl.  
...  

```

Control Font

Since the Tool-Bar control is a container, the controls it contains may inherit some peculiarities.

The font handle set to this property will be the default value of the FONT Property of the controls created in

the Tool-Bar control.

This property also affects the way the cell size is calculated. See the [Cell Height](#), [Cell Size](#), and [Cell Width](#) properties for further details.

Example - Define a tool-bar with 3 buttons and default control font

```
screen section.
...
01 screen-1-tb-1.
03 screen-1-pb-4 Push-Button
   line 1.7
   column 2.4
   size 5.2 cells
   lines 2.6 cells
   id 11
   title "Exit"
   .
03 screen-1-pb-5 Push-Button
   line 1.7
   column 13.0
   size 5.2 cells
   lines 2.6 cells
   id 12
   title ">"
   .
03 screen-1-pb-6 Push-Button
   line 1.7
   column 20.1
   size 5.2 cells
   lines 2.6 cells
   id 13
   title "<"
   .
...
procedure division.
...
   display tool-bar
      cell width 10
      cell height 10
      control font Default-Font
      color 232
      lines 4.0
      handle screen-1-tb-1-hdl
   .
   display screen-1-tb-1 upon screen-1-tb-1-hdl.
...
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Display a tool-bar with a base css style

```
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        color 232  
        lines 4.0  
        css-base-style-name "css-tool-bar"  
        handle screen-1-tb-1-hdl
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Display a tool-bar with a css style

```
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        color 232  
        lines 4.0  
        css-style-name "css-tool-bar"  
        handle screen-1-tb-1-hdl
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a tool-bar

```
procedure division.  
...  
    modify screen-1-tb-1 custom-data "Screen 1 custom data"  
...
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Tool-Bar control.

The value set here is also the default foreground color of the controls created in the Tool-Bar control.

See "[Color management](#)" for further details.

Example - Define a tool-bar with 3 buttons and foreground and background color

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
        .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
        .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
        .  
...  
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        background-color 6  
        foreground-color 7  
        lines 4.0  
        handle screen-1-tb-1-hdl  
        .  
    display screen-1-tb-1 upon screen-1-tb-1-hdl.  
...
```

Help-Id

This property is ignored by the Tool-Bar control.

Hint

This property is ignored by the Tool-Bar control.

Id

This property is ignored by the Tool-Bar control.

Layout-Manager

This property associates a Layout Manager with the Tool-Bar. Three types of Layout Managers (defined in the [isresize.def](#) Copybook) are supported.

- LM-RESIZE
- LM-RESPONSIVE
- LM-SCALE

Example - Define a tool-bar with 3 buttons and layout-manager

```
screen section.  
...  
01 screen-1-tb-1.  
03 screen-1-pb-4 Push-Button  
   line 1.7  
   column 2.4  
   size 5.2 cells  
   lines 2.6 cells  
   id 11  
   title "Exit"  
   .  
03 screen-1-pb-5 Push-Button  
   line 1.7  
   column 13.0  
   size 5.2 cells  
   lines 2.6 cells  
   id 12  
   title ">"  
   .  
03 screen-1-pb-6 Push-Button  
   line 1.7  
   column 20.1  
   size 5.2 cells  
   lines 2.6 cells  
   id 13  
   title "<"  
   .  
...  
procedure division.  
...  
display tool-bar  
   cell width 10  
   cell height 10  
   control font Default-Font  
   color 232  
   lines 4.0  
   layout-manager lm-resize  
   handle screen-1-tb-1-hdl  
   .
```

Lines

This property allows you to specify the height of the Tool-Bar control. Decimal values are allowed.

Example - Define a tool-bar with 3 buttons and lines 4

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
    .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
    .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
    .  
...  
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        color 232  
        lines 4.0  
        layout-manager lm-resize  
        handle screen-1-tb-1-hdl  
    .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Tool-Bar control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Display a tool-bar with a pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
procedure division.  
...  
    display tool-bar  
        cell width 10  
        cell height 10  
        control font Default-Font  
        color 232  
        lines 4.0  
        pop-up menu hmenu  
        handle screen-1-tb-1-hdl  
    .
```

Styles

The following styles are applicable to the TOOL-BAR control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [High](#), [Highlight](#), [Low](#), [Lowlight](#), [Moveable](#), [Multiline](#), [Standard](#).

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a tool-bar with 3 buttons and high background and bold foreground

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
        .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
        .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
        .  
...  
procedure division.  
...  
    display tool-bar  
        background-high  
        bold  
        cell width 10  
        cell height 10  
        control font Default-Font  
        color 232  
        lines 4.0  
        layout-manager lm-resize  
        handle screen-1-tb-1-hdl  
        .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a tool-bar with 3 buttons and high background and bold foreground

```
screen section.
...
01 screen-1-tb-1.
    03 screen-1-pb-4 Push-Button
        line 1.7
        column 2.4
        size 5.2 cells
        lines 2.6 cells
        id 11
        title "Exit"
        .
    03 screen-1-pb-5 Push-Button
        line 1.7
        column 13.0
        size 5.2 cells
        lines 2.6 cells
        id 12
        title ">"
        .
    03 screen-1-pb-6 Push-Button
        line 1.7
        column 20.1
        size 5.2 cells
        lines 2.6 cells
        id 13
        title "<"
        .
...
procedure division.
...
    display tool-bar
        background-high
        bold
        cell width 10
        cell height 10
        control font Default-Font
        color 232
        lines 4.0
        layout-manager lm-resize
        handle screen-1-tb-1-hdl
        .
```

Moveable

When this style is set, the user can detach the Tool-Bar control from its window.

Example - Define a tool-bar with 3 buttons and moveable style

```
screen section.  
...  
01 screen-1-tb-1.  
    03 screen-1-pb-4 Push-Button  
        line 1.7  
        column 2.4  
        size 5.2 cells  
        lines 2.6 cells  
        id 11  
        title "Exit"  
        .  
    03 screen-1-pb-5 Push-Button  
        line 1.7  
        column 13.0  
        size 5.2 cells  
        lines 2.6 cells  
        id 12  
        title ">"  
        .  
    03 screen-1-pb-6 Push-Button  
        line 1.7  
        column 20.1  
        size 5.2 cells  
        lines 2.6 cells  
        id 13  
        title "<"  
        .  
...  
procedure division.  
...  
    display tool-bar moveable  
        background-high  
        bold  
        cell width 10  
        cell height 10  
        control font Default-Font  
        color 232  
        lines 4.0  
        layout-manager lm-resize  
        handle screen-1-tb-1-hdl  
        .
```

Multiline

When this style is set, controls can be displayed on multiple lines in the Tool-Bar. Without this style, by default, controls are vertically centered in the Tool-Bar area.

Example - Define a tool-bar with 3 buttons and multiline style

```
screen section.  
...  
01 screen-1-tb-1.  
03 screen-1-pb-4 Push-Button  
   line 1.7  
   column 2.4  
   size 5.2 cells  
   lines 2.6 cells  
   id 11  
   title "Exit"  
   .  
03 screen-1-pb-5 Push-Button  
   line 1.7  
   column 13.0  
   size 5.2 cells  
   lines 2.6 cells  
   id 12  
   title ">"  
   .  
03 screen-1-pb-6 Push-Button  
   line 1.7  
   column 20.1  
   size 5.2 cells  
   lines 2.6 cells  
   id 13  
   title "<"  
   .  
...  
procedure division.  
...  
display tool-bar moveable multiline  
background-high  
bold  
cell width 10  
cell height 10  
control font Default-Font  
color 232  
lines 4.0  
layout-manager lm-resize  
handle screen-1-tb-1-hdl  
.
```

Events

The following events are applicable to the TOOL-BAR control: [MSG-END-MENU](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#).

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

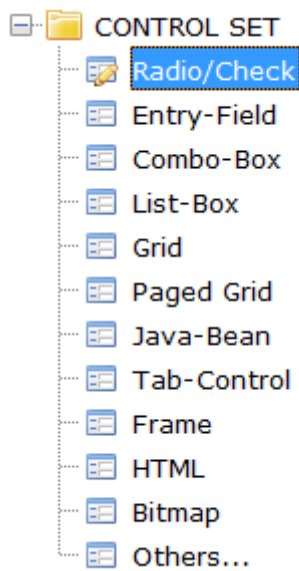
Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

TREE-VIEW



A Tree-View can represent hierarchical structures. It contains items that can have one or more children. Each item can contain graphics, text or both. Item text can also be editable.

Properties

The following properties are applicable to the TREE-VIEW control: [Action](#), [Background-Color](#), [Bitmap-Handle](#), [Bitmap-Number](#), [Bitmap-Width](#), [Border-Color](#), [Col](#), [Color](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Ensure-Visible](#), [Event-List](#), [Exclude-Event-List](#), [Expand](#), [Font](#), [Foreground-Color](#), [Has-Children](#), [Help-Id](#), [Hidden-Data](#), [Hint](#), [Id](#), [Item](#), [Item-Hint](#), [Item-Text](#), [Item-To-Add](#), [Item-To-Delete](#), [Item-To-Empty](#), [Layout-data](#), [Line](#), [Lines](#), [Mass-Update](#), [Max-Height](#), [Max-Width](#), [Min-Height](#), [Min-Width](#), [Next-Item](#), [Parent](#), [Placement](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Reset-List](#), [Size](#), [Value](#), [Visible](#).

Action

A specific action is performed when a value is assigned to this property. Only the following symbolic value,

included in the copy file [isgui.def](#), can be assigned. The table below shows the action related to it:

action-entry	The Tree-View control enters the Edit Mode and the MSG-BEGIN-ENTRY event is generated.
action-end	The Tree-View control stops the edit mode and the MSG-FINISH-ENTRY event is generated.

Example - Define a tree-view and use the action property in procedure division to start the edition of the current item

```
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 23.6 cells
   lines 29.1 cells
   id 2
   no-box
   .
...
procedure division.
...
   modify screen-1-tv-1
      parent = null
      item-to-add "Option 1"
      giving tv-item-hdl
      parent = tv-item-hdl
      item-to-add "Sub-option 1.1"
      giving tv-item-hdl2
      parent = tv-item-hdl
      item-to-add "Sub-option 1.2"
      giving tv-item-hdl3
      parent = null
      item-to-add "Option 2"
      giving tv-item-hdl0
      parent = tv-item-hdl0
      item-to-add "Sub-option 2.1"
      giving tv-item-hdl4
      parent = null
      item-to-add "Option 3"
      giving tv-item-hdl1
      .
...
   modify screen-1-tv-1 action action-entry
...

```

Background-Color

This property allows you to set or retrieve the background color of the Tree-View control. See "[Color management](#)" for further details.

Example - Define a tree-view with background and foreground color

```
screen section.  
...  
    03 screen-1-tv-1 Tree-View  
        line 2.7  
        column 3.4  
        size 23.6 cells  
        lines 29.1 cells  
        background-color 6  
        foreground-color 7  
        id 2  
        no-box  
        .
```

Bitmap-Handle

This property identifies the bitmap strip handle to be used in the Tree-View.

Example - Define a tree-view that uses bitmaps for the item icons

```
working-storage section.
77 app-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-tv-1 Tree-View
  line 2.7
  column 3.4
  size 20.8 cells
  lines 29.1 cells
  background-color 6
  foreground-color 7
  id 2
  no-box
  bitmap-handle app-png
  bitmap-width 20
  .
...
procedure division.
...
  call "w$bitmap" using wbitmap-load "app.png" giving app-png.
...
  modify screen-1-tv-1
    parent = null
    item-to-add "Option 1"
    giving tv-item-hdl
    bitmap-number 1
    parent = tv-item-hdl
    item-to-add "Sub-option 1.1"
    giving tv-item-hdl2
    bitmap-number 2
    parent = tv-item-hdl
    item-to-add "Sub-option 1.2"
    giving tv-item-hdl3
    bitmap-number 2
    parent = null
    item-to-add "Option 2"
    giving tv-item-hdl0
    bitmap-number 1
    parent = tv-item-hdl0
    item-to-add "Sub-option 2.1"
    giving tv-item-hdl4
    bitmap-number 2
    parent = null
    item-to-add "Option 3"
    giving tv-item-hdl1
    bitmap-number 1
    .
  .
...
```

Bitmap-Number

This property identifies which bitmap (among the ones in the bitmap strip referenced by the [Bitmap-Handle](#) property) is to be displayed in the item identified by the [Item](#) property.

Example - Define a tree-view that uses bitmaps for the item icons

```
working-storage section.
77 app-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-tv-1 Tree-View
  line 2.7
  column 3.4
  size 20.8 cells
  lines 29.1 cells
  background-color 6
  foreground-color 7
  id 2
  no-box
  bitmap-handle app-png
  bitmap-width 20
  .
...
procedure division.
...
  call "w$bitmap" using wbitmap-load "app.png" giving app-png.
...
  modify screen-1-tv-1
    parent = null
    item-to-add "Option 1"
    giving tv-item-hdl
    bitmap-number 1
    parent = tv-item-hdl
    item-to-add "Sub-option 1.1"
    giving tv-item-hdl2
    bitmap-number 2
    parent = tv-item-hdl
    item-to-add "Sub-option 1.2"
    giving tv-item-hdl3
    bitmap-number 2
    parent = null
    item-to-add "Option 2"
    giving tv-item-hdl0
    bitmap-number 1
    parent = tv-item-hdl0
    item-to-add "Sub-option 2.1"
    giving tv-item-hdl4
    bitmap-number 2
    parent = null
    item-to-add "Option 3"
    giving tv-item-hdl1
    bitmap-number 1
    .
  .
...
```

Bitmap-Width

This property identifies the width in pixels of the image displayed in the Tree-View. The bitmap strip identified by the [Bitmap-Handle](#) property is divided into several smaller images. The width of each image is the value assigned to this property.

Example - Define a tree-view that uses bitmaps for the item icons

```
working-storage section.
77 app-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-tv-1 Tree-View
  line 2.7
  column 3.4
  size 20.8 cells
  lines 29.1 cells
  background-color 6
  foreground-color 7
  id 2
  no-box
  bitmap-handle app-png
  bitmap-width 20
  .
...
procedure division.
...
  call "w$bitmap" using wbitmap-load "app.png" giving app-png.
...
  modify screen-1-tv-1
    parent = null
    item-to-add "Option 1"
    giving tv-item-hdl
    bitmap-number 1
    parent = tv-item-hdl
    item-to-add "Sub-option 1.1"
    giving tv-item-hdl2
    bitmap-number 2
    parent = tv-item-hdl
    item-to-add "Sub-option 1.2"
    giving tv-item-hdl3
    bitmap-number 2
    parent = null
    item-to-add "Option 2"
    giving tv-item-hdl0
    bitmap-number 1
    parent = tv-item-hdl0
    item-to-add "Sub-option 2.1"
    giving tv-item-hdl4
    bitmap-number 2
    parent = null
    item-to-add "Option 3"
    giving tv-item-hdl1
    bitmap-number 1
    .
  .
...
```

Border-Color

This property allows you to set or retrieve the border color of the Tree-View control. See ["Color management"](#) for further details. The border color is applicable only to controls with the [Boxed](#) style.

Example - Define a tree-view with red border

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   border-color 5  
   id 2  
   .  
...
```

[Col | Column | Pos | Position]

This property allows you to specify the Tree-View control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Tree-View control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Tree-View control will be relative to the ending position of the prior Screen Section item.

When the Tree-View control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).  
03 Tree-View, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a tree-view at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .  
...
```

Color

This property allows you to set or retrieve the color of the Tree-View control. Foreground and background color values are combined and therefore RGB colors are not supported. See "[Color management](#)" for further details.

Example - Define a tree-view with color property

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   id 2  
   no-box  
   bitmap-handle app-png  
   bitmap-width 20  
   .
```

Css-Base-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a tree-view with css-base-style-name, applicable with WD2

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   id 2  
   css-base-style-name "css-tree-view"  
   no-box  
   bitmap-handle app-png  
   bitmap-width 20  
   .
```

Css-Style-Name

This property only has an effect in a Web Direct 2.0 environment. See [Customize the EIS WD2 Layout using CSS](#) for details.

Example - Define a tree-view with css-style-name, applicable with WD2

```
screen section.  
...  
  03 screen-1-tv-1 Tree-View  
    line 2.7  
    column 3.4  
    size 20.8 cells  
    lines 29.1 cells  
    color 144  
    id 2  
    css-style-name "css-tree-view"  
    no-box  
    bitmap-handle app-png  
    bitmap-width 20  
    .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a tree-view

```
procedure division.  
...  
  modify screen-1-tv-1 custom-data "Screen 1 custom data"  
...
```

Enabled

This property assumes a value of "0" if the Tree-View control is disabled, "1" if it is enabled.

Example - Define a tree-view initially disabled and enable it later

```
screen section.  
...  
  03 screen-1-tv-1 Tree-View  
    line 2.7  
    column 3.4  
    size 20.8 cells  
    lines 29.1 cells  
    color 144  
    enabled 0  
    id 2  
    no-box  
    .  
...  
procedure division.  
...  
  modify screen-1-tv-1 enable 1  
...
```

Ensure-Visible

When set to a valid item, that item is made visible. If necessary, items will be expanded and the content of the control will scroll.

Example - Define a tree-view and ensure one of its subitems is visible all the time

```
working-storage section.
77 app-png pic s9(9) comp-4.
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   id 2
   no-box
   bitmap-handle app-png
   bitmap-width 20
   .
...
procedure division.
...
   call "w$bitmap" using wbitmap-load "app.png" giving app-png.
...
   modify screen-1-tv-1
       parent = null
       item-to-add "Option 1"
       giving tv-item-hdl
       bitmap-number 1
       parent = tv-item-hdl
       item-to-add "Sub-option 1.1"
       giving tv-item-hdl2
       bitmap-number 2
       parent = tv-item-hdl
       item-to-add "Sub-option 1.2"
       giving tv-item-hdl3
       bitmap-number 2
       parent = null
       item-to-add "Option 2"
       giving tv-item-hdl0
       bitmap-number 1
       parent = tv-item-hdl0
       item-to-add "Sub-option 2.1"
       giving tv-item-hdl4
       bitmap-number 2
       ensure-visible = tv-item-hdl4
       parent = null
       item-to-add "Option 3"
       giving tv-item-hdl1
       bitmap-number 1
       .
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a tree-view with a excluded event list

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   id 2  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   no-box  
   .
```

Exclude-Event-List

If this property is set to "1", then none of the events listed in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a tree-view with a excluded event list

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   id 2  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   no-box  
   .
```

Expand

This property is used to programmatically expand or collapse the item identified by the [Item](#) property. Valid

values, defined in [isgui.def](#), are:

tvflag-collapse	The item is collapsed.
tvflag-expand	The item is expanded.

Example - Define a tree-view and expand one of its items on procedure division

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   id 2  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   no-box  
   .  
...  
procedure division.  
...  
   modify screen-1-tv-1 item 1 expand tvflag-expand  
...  

```

Font

This property specifies the font used to display the content of the Tree-View control. It may be used to compute the height and the width of the Tree-View control, as well. See the [Height-In-Cells](#), [Lines](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a tree-view with special font

```
working-storage section.  
77 Tahoma-10v0 handle of font.  
...  
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   font Tahoma-10v0  
   id 2  
   no-box  
   .  
...  
procedure division.  
...  
   initialize wfont-data tahoma-10v0.  
   move 10 to wfont-size.  
   move "Tahoma" to wfont-name.  
   set wfont-bold to false.  
   set wfont-italic to false.  
   set wfont-underline to false.  
   set wfont-strikeout to false.  
   set wfont-fixed-pitch to false.  
   call "w$font" using wfont-get-font tahoma-10v0 wfont-data.  
...  

```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Tree-View control. See "[Color management](#)" for further details.

Example - Define a tree-view with background and foreground color

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 23.6 cells  
   lines 29.1 cells  
   background-color 6  
   foreground-color 7  
   id 2  
   no-box  
   .  

```

Has-Children

This property is used to set or inquire whether the item identified by the [Item](#) property has children.

When set to a non-zero value, the item is marked as if it had children, even if it does not actually contain any item. This way, when the user tries to expand it, the [MSG-TV-EXPANDING](#) and [MSG-TV-EXPANDED](#) events are fired. This technique is used to handle a Tree-View control with items loaded on demand.

When an item is added, the HAS-CHILDREN property of its parent is automatically set to 1.

When an item is emptied, using the [Item-To-Empty](#) property, its HAS-CHILDREN property is automatically set to 0.

Example - Add a parent item to the tree-view specifying that it will have children. Later, check if that item has children

```
working-storage section.  
77 w-item handle.  
77 hc-flg pic 9.  
...  
procedure division.  
...  
    modify screen-1-tv-1 parent 0  
        item-to-add "Parent" giving w-item  
        has-children 1  
...  
    inquire screen-1-tv-1(w-item) has-children hc-flg
```

Help-Id

This property allows you to assign a unique ID to the Tree-View control to be passed to the help processor.

See [Help automation](#) for more information.

Example - Define a tree-view with help-id

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
    line 2.7  
    column 3.4  
    size 20.8 cells  
    lines 29.1 cells  
    color 144  
    help-id 4433  
    id 2  
    no-box  
    .
```

Hidden-Data

This property allows you to set or retrieve the hidden value of the item identified by the [Item](#) property. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to an item.

Example - Modify a tree-view and set the hidden-data to a tree-view to a program name to be called

```
...  
procedure division.  
...  
    modify screen-1-tv-1  
        parent = null  
        item-to-add "Option 1"  
        giving tv-item-hdl  
        bitmap-number 1  
        parent = tv-item-hdl  
        item-to-add "Sub-option 1.1"  
        giving tv-item-hdl2  
        bitmap-number 2  
        parent = tv-item-hdl  
        item-to-add "Sub-option 1.2"  
        giving tv-item-hdl3  
        bitmap-number 2  
        parent = null  
        item-to-add "Option 2"  
        giving tv-item-hdl0  
        bitmap-number 1  
        parent = tv-item-hdl0  
        item-to-add "Sub-option 2.1"  
        giving tv-item-hdl4  
        bitmap-number 2  
        ensure-visible = tv-item-hdl4  
        parent = null  
        item-to-add "Option 3"  
        giving tv-item-hdl1  
        bitmap-number 1  
        hidden-data "PROG003"  
        .
```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Tree-View control.

Example - Define a tree-view with a hint text

```
screen section.  
...  
    03 screen-1-tv-1 Tree-View  
        line 2.7  
        column 3.4  
        size 20.8 cells  
        lines 29.1 cells  
        color 144  
        id 2  
        hint "Main Menu"  
        no-box  
        .
```

Id

This property allows you to assign a unique ID to the Tree-View control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a tree-view with an ID

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   id 2  
   no-box  
   .
```

Item

This property represents the item that the following properties refer to: [Bitmap-Number](#), [Expand](#), [Has-Children](#), [Hidden-Data](#), [Item-Text](#) and [Next-Item](#).

Example - Modify a tree-view item to expand it using the item property

```
...  
procedure division.  
...  
   modify screen-1-tv-1 item 1 expand tvflag-expand  
...  

```

Item-Hint

This alphanumeric property is used to set or retrieve the hint text of an item identified by the [Item](#) property. Such text is shown when the mouse pointer is placed on the Tree-View item.

Example - Modify the hint text of a specific item of a tree-view

```
...  
procedure division.  
...  
   modify screen-1-tv-1 item 1 item-hint "Hint of item 1"  
...  

```

Item-Text

This alphanumeric property is used to set or retrieve the text of an item identified by the [Item](#) property.

Example - Modify the item text of a tree-view item

```
...  
procedure division.  
...  
    modify screen-1-tv-1 item 1 item-text "Text of Option 1"  
...  

```

Item-To-Add

When a value is assigned to this property, a new item is added to the Tree-View control.

Multiple values can be added at the same time, enclosed between parentheses.

Each new item will be added according to the values set to the [Parent](#) and [Placement](#) properties.

Example - Add a new item to the tree-view

```
...  
working-storage section.  
77 tv-item-hdl1 handle.  
...  
procedure division.  
...  
    modify screen-1-tv-1  
        parent = null  
        item-to-add "Option 1"  
        giving tv-item-hdl  
...  

```

Item-To-Delete

As soon as the value of this property is modified, the corresponding item is removed.

Each item has a unique ID that is automatically assigned at the moment it is created.

Example - Modify a tree-view to delete the 3rd item

```
...  
procedure division.  
...  
    modify screen-1-tv-1 item-to-delete 3  
...  

```

Item-To-Empty

As soon as the value of this property is modified, the corresponding item is emptied. The item itself is not deleted, only its children are removed from the Tree-View control.

Each item has a unique ID that is automatically assigned at the moment it is created.

Example - Modify a tree-view an empty one of its items

```
...  
procedure division.  
...  
    modify screen-1-tv-1 item-to-empty 1  
...
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a tree-view with a layout-data to resize in X and Y when the layout manager requires so

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
    line 2.7  
    column 3.4  
    size 20.8 cells  
    lines 29.1 cells  
    color 144  
    id 2  
    layout-data 17  
    no-box  
    .
```

Line

This property allows you to specify the Tree-View control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Tree-View control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Tree-View control will be relative to the starting position of the prior Screen Section item.

When the Tree-View control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Tree-View, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a tree-view at line 8.0 on the screen section definition

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
    line 8.0  
    column 5.0  
    color 7  
    size 45.0 cells  
    id 2  
    .
```

Lines

This property allows you to specify the height of the Tree-View control. If the **PIXEL** keyword follows the value specified here, the height is computed in pixels. If either the **CELLS** keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Tree-View control is still computed in **CELLS**, but the cell size is based on the font set for the Tree-View control with the [Font](#) property. If no font has been defined for the Tree-View control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a tree-view with height in lines

```
...  
screen section.  
...  
03 screen-1-tv-1 Tree-View  
    line 2.7  
    column 3.4  
    size 20.8 cells  
    lines 29.1 cells  
    color 144  
    font Tahoma-10v0  
    id 2  
    .
```

Mass-Update

Setting this property to "1" keeps the isCOBOL framework from repainting the Tree-View control every time the program modifies it. This practice is recommended to increase performance when a large number of changes are applied to the Tree-View control. At the end of the process it is necessary to reset the property to its default value "0" to see the changes.

Example - Load three items in mass update mode

```
modify screen-1-tv-1 mass-update = 1  
modify screen-1-tv-1 parent null, item-to-add "item-1"  
modify screen-1-tv-1 parent null, item-to-add "item-2"  
modify screen-1-tv-1 parent null, item-to-add "item-3"  
modify screen-1-tv-1 mass-update = 0
```


Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a tree-view with maximum and minimum dimensions for the layout manager

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   id 2
   max-width 40.0
   min-width 10.0
   min-height 15.0
   max-height 50.0
   layout-data 17
   .
...
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a tree-view with maximum and minimum dimensions for the layout manager

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   id 2
   max-width 40.0
   min-width 10.0
   min-height 15.0
   max-height 50.0
   layout-data 17
   .
...
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a tree-view with maximum and minimum dimensions for the layout manager

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   id 2
   max-width 40.0
   min-width 10.0
   min-height 15.0
   max-height 50.0
   layout-data 17
   .
...
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a tree-view with maximum and minimum dimensions for the layout manager

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   id 2
   max-width 40.0
   min-width 10.0
   min-height 15.0
   max-height 50.0
   layout-data 17
   .
...
```

Next-Item

This property returns an item ID. The value set to the property is the item whose ID is needed. It may refer to

the whole Tree-View control or to another item. Valid values, defined in `isgui.def`, are:

tvni-child	<p>The first child of the item identified by the <code>Item</code> property is returned. If it has no children, zero is returned. This does not mean that the item cannot be expanded, an item can be expanded when its <code>Has-Children</code> property is set to a non-zero value.</p> <pre> MODIFY MY_TREE-VIEW, ITEM = AnyItem NEXT-ITEM = tvni-child GIVING FirstChildItemID </pre>
tvni-first-visible	<p>The id of the first visible item in the Tree-View control is returned.</p> <pre> MODIFY MY_TREE-VIEW, NEXT-ITEM = tvni-first-visible GIVING FirstVisibleItemID </pre>
tvni-next	<p>The next sibling of the item identified by the <code>Item</code> property is returned. If the item identified by the <code>Item</code> property is the last of its level, zero is returned.</p> <pre> MODIFY MY_TREE-VIEW, ITEM = AnyItem NEXT-ITEM = tvni-next GIVING NextItemID </pre>
tvni-next-visible	<p>The next visible item after the item identified by the <code>Item</code> property is returned. The <code>Item</code> property must refer to a visible item.</p> <pre> MODIFY MY_TREE-VIEW, ITEM = AnyItem NEXT-ITEM = tvni-next-visible GIVING NextVisibleItemID </pre>
tvni-parent	<p>The parent of the item identified by the <code>Item</code> property is returned. If the item identified by the <code>Item</code> property is at the top level, zero is returned.</p> <pre> MODIFY MY_TREE-VIEW, ITEM = AnyItem NEXT-ITEM = tvni-parent GIVING ParentItemID </pre>
tvni-previous	<p>The previous sibling of the item identified by the <code>Item</code> property is returned. If the item identified by the <code>Item</code> property is the first of its level, zero is returned.</p> <pre> MODIFY MY_TREE-VIEW, ITEM = AnyItem NEXT-ITEM = tvni-previous GIVING PreviousItemID </pre>
tvni-previous-visible	<p>The previous visible item before the item identified by the <code>Item</code> property is returned. The <code>Item</code> property must refer to a visible item.</p> <pre> MODIFY MY_TREE-VIEW, ITEM = AnyItem NEXT-ITEM = tvni-previous-visible GIVING PreviousVisibleItemID </pre>
tvni-root	<p>The id of the very first item in the Tree-View control is returned.</p> <pre> MODIFY MY_TREE-VIEW, NEXT-ITEM = tvni-root GIVING RootItemID </pre>

Parent

The value set to this property affects the place where the next item will be added the next time the [Item-To-Add](#) property will be set. When set to zero, the new item will be added at the top level.

Items with the same parent belong to the same level and are called siblings.

Setting this property after the [Item-To-Add](#) property is not recommended because it may lead to unexpected behaviors. A correct code will always look as the sample below.

Example - Add a child item to a given parent item

```
modify screen-1-tv-1,  
  parent      = parentItemId,  
  item-to-add = "new item"
```

Placement

The value set to this property affects the place where the next item will be added the next time the [Item-To-Add](#) property will be set. It can be one of the following values:

tvplace-first	The new item will be the first item of its level.
tvplace-last	The new item will be the last item of its level.
tvplace-sort	The items in the level will be sorted. It does not make sense to set tvplace-sort for a level where items are not sorted. In other words, if at least one item is added to the level with the PLACEMENT property set to tvplace-sort, all the other items of the same level should be added the same way.
<i>AnyltemID</i>	The new item will be placed after the item identified by AnyltemID. Obviously, AnyltemID must refer to an item of the level where the new item will be inserted.

Setting this property after the [Item-To-Add](#) property is not recommended because it may lead to unexpected behaviors. A correct code will always look as follows:

```
modify screen-1-tv-1,  
  parent      = parentItemId,  
  placement    = tvplace-sort,  
  item-to-add = "new item"
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Tree-View control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Define a tree-view with pop-up menu

```
working-storage section.  
77 hmenu pic s9(9) comp-4.  
...  
screen section.  
...  
    03 screen-1-tv-1 Tree-View  
        pop-up menu hmenu  
        line 32.5  
        column 5.2  
        size 23.3 cells  
        lines 11.6 cells  
        id 9  
        no-box  
        .  
...  
*> Use w$menu in procedure division to build the pop-up menu  
...
```

Reset-List

By assigning a value other than zero to this property, all the items are removed from the Tree-View control.

Example - Modify a tree-view to reset its contents

```
...  
procedure division.  
...  
    modify screen-1-tv-1 reset-list 1  
...  

```

Size

This property allows you to specify the size of the Tree-View control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the **Width-In-Cells** style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the **Width-In-Cells** style is not set, the size of the Tree-View control is still computed in CELLS, but the cell size is based on the font set for the Tree-View control with the **Font** property. If no font has been defined for the Tree-View control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a tree-view with size

```
...  
screen section.  
...  
  03 screen-1-tv-1 Tree-View  
    line 2.7  
    column 3.4  
    size 20.8 cells  
    lines 29.1 cells  
    color 144  
    id 2  
    .
```

Value

This property represents the value of the Tree-View control.

When inquired, it returns the value that is currently represented.

When set, the Tree-View control changes its look to represent it.

It is the currently selected item.

Example - Get the value of the currently selected item from the tree-view

```
working-storage section.  
77 ws-tv    pic 9(3).  
...  
procedure division.  
...  
  inquire screen-1-tv-1 value ws-tv  
...  

```

Visible

This property assumes a value of "0" if the Tree-View control is not visible, "1" if it is visible.

Example - Make invisible a tree-view

```
...  
procedure division.  
...  
  modify screen-1-tv-1 visible 0
```

Styles

The following styles are applicable to the TREE-VIEW control: 3-D, Background-High, Background-Low, Background-Standard, Bold, Boxed, Buttons, Height-In-Cells, High, Highlight, Lines-At-Root, Low, Lowlight, No-Box, Permanent, Show-Lines, Show-Sel-Always, Standard, Temporary, Width-In-Cells.

{ 3-D | Boxed | No-Box }

3-D	The box drawn around the Tree-View control appears with a 3-D effect.
Boxed	A flat box is drawn around the Tree-View control.
No-Box	No box is drawn around the Tree-View control. Set this style when you need to save space.

The visual result may vary with different Swing LAF (Look And Feel).

Example - Define a 3D tree-view

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   id 2
   3-d
   .
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a tree-view with background high and bold foreground

```
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   color 144
   font Tahoma-10v0
   id 2
   background-high
   bold
   .
```

Buttons

This keyword is supported for compatibility but it has no specific effect. In order to see buttons along with the Tree-View items, you must specify either the [Lines-At-Root](#) or [Show-Lines](#) styles. The button aspect is influenced by the Look & Feel. Items with children can be expanded even without buttons, by double clicking on their text.

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Tree-View control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines](#) value CELLS".

Example - Define a tree-view with height in cells

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1  
   height-in-cells  
   color 144  
   id 2  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a tree-view with background high and bold foreground

```
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8 cells  
   lines 29.1 cells  
   color 144  
   font Tahoma-10v0  
   id 2  
   background-high  
   bold  
   .
```


Lines-At-Root

When this style is set, the lines connecting the items at the same level are shown also for top level items and, if they have children, a button is shown before their name. This makes the Tree-View control more readable, without influencing its behavior. The lines may not show up using certain Look & Feel.

Example - Define a tree-view with lines-at-root style

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   height-in-cells
   color 144
   id 2
   3-d
   lines-at-root
   .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a temporary tree-view

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   height-in-cells
   color 144
   id 2
   3-d
   temporary
   .
```

Show-Lines

When this style is set, items at the same level are connected by a line and, if they have children, a button is shown before their name. This just makes the Tree-View control more readable, without influencing its behavior. The lines may not show up using certain Look & Feel.

Example - Define a tree-view with show-lines style

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   height-in-cells
   color 144
   id 2
   show-lines
   .
```

Show-Sel-Always

When this style is set, the current item is highlighted even if the Tree-View control is not the current control.

When this style is not set, there is no way to distinguish the current item if the Tree-View control is not the current control. This is the default setting.

Example - Define a tree-view with show-sel-always style

```
...
screen section.
...
03 screen-1-tv-1 Tree-View
   line 2.7
   column 3.4
   size 20.8 cells
   lines 29.1 cells
   height-in-cells
   color 144
   id 2
   show-sel-always
   show-lines
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Tree-View control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a tree-view with width-in-cells

```
...  
screen section.  
...  
03 screen-1-tv-1 Tree-View  
   line 2.7  
   column 3.4  
   size 20.8  
   lines 29.1 cells  
   height-in-cells  
   color 144  
   id 2  
   show-sel-always  
   show-lines  
   width-in-cells  
   .
```

Events

The following events are applicable to the TREE-VIEW control: [CMD-GOTO](#), [CMD-HELP](#), [MSG-BEGIN-ENTRY](#), [MSG-CANCEL-ENTRY](#), [MSG-END-MENU](#), [MSG-FINISH-ENTRY](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [MSG-TV-DBLCLICK](#), [MSG-TV-EXPANDED](#), [MSG-TV-EXPANDING](#), [MSG-TV-OUT-NEXT](#), [MSG-TV-OUT-PREV](#), [MSG-TV-SELCHANGE](#), [MSG-TV-SELCHANGING](#), [MSG-VALIDATE](#).

CMD-GOTO

This event is fired when the user tries to activate the Tree-View control with the mouse or by pressing the associated key letter.

CMD-HELP

This event is fired when the help for the Tree-View control is requested. The EVENT-DATA-2 data item contains the [Help-Id](#) for the Tree-View control.

MSG-BEGIN-ENTRY

This event is fired when the user starts changing the text of an item. The EVENT-DATA-2 data item contains the ID of the item being changed. The [Item](#) property is automatically set to the value of that item. Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the data item from being edited.

MSG-CANCEL-ENTRY

This event is fired when the user is changing the text of an item and presses the [Escape] key. The EVENT-DATA-2 data item contains the ID of the current item. The [Item](#) property is automatically set to the value of that item. At the end of the Event Procedure, the original text is restored.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-FINISH-ENTRY

This event is fired when the user finishes changing the text of an item. The EVENT-DATA-2 data item contains the ID of the current item. The [Item](#) property is automatically set to the value of that item. Setting EVENT-ACTION to EVENT-ACTION-FAIL keeps the item in edit mode.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

MSG-TV-DBLCLICK

This event is fired when the user double-clicks a tree-view item without children. If the item has children, the [MSG-TV-EXPANDING](#) and the [MSG-TV-EXPANDED](#) events are fired instead. EVENT-DATA-2 contains the ID of the expanded item.

MSG-TV-EXPANDED

This event is fired after a tree-view item has expanded or collapsed as a consequence of the user's double-clicking a tree-view item with children. EVENT-DATA-1 contains the symbolic values TVFLAG-EXPAND or TVFLAG-COLLAPSE depending on the action performed. EVENT-DATA-2 contains the ID of the expanded item. Items can be expanded or collapsed by clicking the [-], [+] or [*] keys, too.

MSG-TV-EXPANDING

This event is fired after a tree-view item is going to be expanded or collapsed as a consequence of the user's double-clicking a tree-view item with children. EVENT-DATA-1 contains the symbolic values TVFLAG-EXPAND or TVFLAG-COLLAPSE depending on the action performed. EVENT-DATA-2 contains the ID of the parent item. Items can be expanded or collapsed by clicking the [-], [+] or [*] keys, too. Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the item from being expanded or collapsed.

MSG-TV-OUT-NEXT

This event is fired when the user press the right arrow key on the children items of the last element with children in a branch or when the user press the down arrow key on the very last item in the Tree-View.

The EVENT-DATA-2 data item contains the ID of the Tree-View item.

MSG-TV-OUT-PREV

This event is fired when the user press the left arrow key on a collapsed item at root level or when the user press the up arrow key on the very first item in the Tree-View.

The EVENT-DATA-2 data item contains the ID of the Tree-View item.

MSG-TV-SELCHANGE

This event is fired when the user changes the selected item in a Tree-View control. The EVENT-DATA-2 data item contains the newly selected item. EVENT-DATA-1 contains one of the following symbolic values, defined in [isgui.def](#), depending on the reason of the change.

TVFLAG-MOUSE	The new item has been selected with the mouse.
TVFLAG-KEYBOARD	The new item has been selected with the keyboard.
TVFLAG-PROGRAM	The new item has been programmatically selected by setting the Value property.

MSG-TV-SELCHANGING

This event is fired when the selected item in a Tree-View control is going to be changed. The EVENT-DATA-2 data item contains the ID of the candidate item. EVENT-DATA-1 contains one of the following symbolic values, defined in [isgui.def](#), depending on the reason of the change.

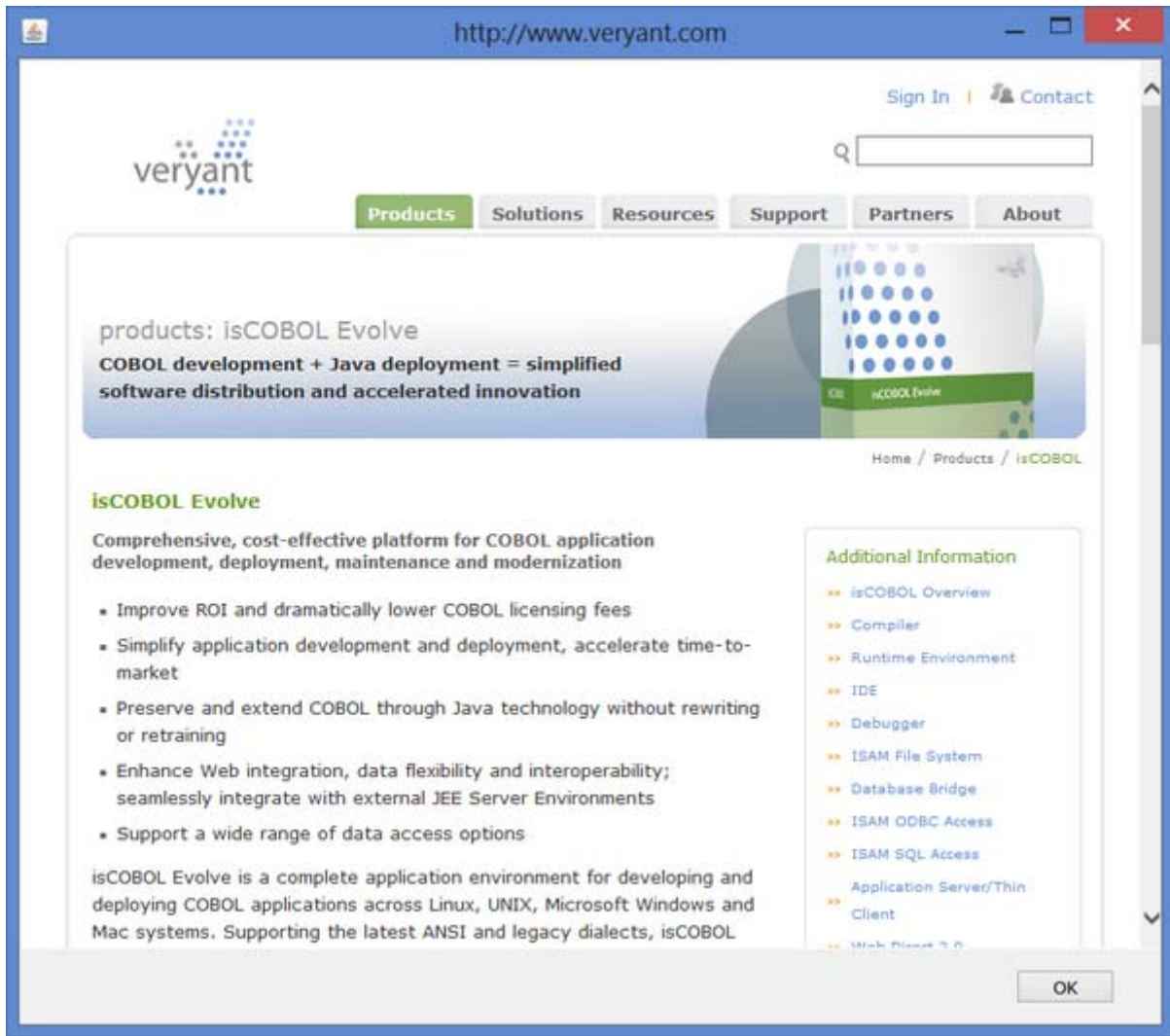
TVFLAG-MOUSE	The new item is going to be selected with the mouse.
TVFLAG-KEYBOARD	The new item is going to be selected with the keyboard.
TVFLAG-PROGRAM	The new item is going to be programmatically selected by setting the Value property.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the new item from being selected.

MSG-VALIDATE

This event is generated when the user transfers the focus to another control with the mouse. Setting EVENT-ACTION to EVENT-ACTION-CONTINUE causes the focus to remain on the control, allowing the user to correct errors.

WEB-BROWSER



A Web-Browser control allows you to embed a native web browser to the screen.

When the focus is on a Web-Browser some keys are trapped by the control and can't be used to interrupt the ACCEPT of the screen. Two typical examples are the ESC key (caught by the browser as "stop loading" command) and the F5 key (caught by the browser as "refresh" command).

The Web-Browser control is implemented through the third party product [DJ](#). Due to rules inherited from this implementation, at the first display of a Web-Browser a peer JVM is created to manage the native part.

If running under Java7 or later, by setting `iscobol.gui.webbrowser.class=com.iscobol.fx.JFXWebBrowser` in the configuration it is possible to use the JavaFX WebView control for the Web-Browser implementation. This approach is preferable in most of cases. See [iscobol.gui.webbrowser.class](#) for details about the features not supported by JavaFX WebView in order to understand if such control is suitable for your application.

Properties

The following properties are applicable to the WEB-BROWSER control: [Busy](#), [Col](#), [Column](#), [Css-Base-Style-Name](#), [Css-Style-Name](#), [Custom-Data](#), [Enabled](#), [Event-List](#), [Exclude-Event-List](#), [File-Name](#), [Font](#), [Go-Back](#), [Go-](#)

Forward, Go-Home, Go-Search, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Progress, Max-Width, Min-Height, Min-Width, Navigate-Url, Pos, Position, Print, Print-No-Prompt, Progress, Refresh, Save-As, Save-As-No-Prompt, Size, Status-Text, Stop-Browser, Title, Value, Visible.

Busy

When inquired, this property returns 1 if the Web-Browser control is still busy loading the URI (Uniform Resource Identifier), otherwise it returns 0.

Example - Define a web-browser control and check if it is busy

```
working-storage section.
...
77 ws-busy pic 9.
...
screen section.
...
03 screen-1-wb-1 Web-Browser
   line 23.1
   column 2.0
   size 62.2 cells
   lines 28.0 cells
   id 15
   value "http://www.veryant.com"
   .
...
procedure division.
...
   inquire screen-1-wb-1 busy ws-busy
   if ws-busy not = 1
       display message "Done loading the URL"
   end-if.
...
```

[Col | Column | Pos | Position]

This property allows you to specify the Web-Browser control's horizontal position. The value is specified in cells. Decimal values are allowed.

When the Web-Browser control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The horizontal position of the Web-Browser control will be relative to the ending position of the prior Screen Section item.

When the Web-Browser control is part of a Screen Section and the COL Property is omitted, COL + 1 is implied.

```
03 Label, COL 2, SIZE 12, (more screen options).
03 Web-Browser, COL + 1, (more screen options).
```

The second control will be placed at column 14.

Example - Position a web-browser at column 5.0 on the screen section definition screen section.

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Css-Base-Style-Name

This property is ignored as the WEB-BROWSER control is not supported in a Web Direct 2.0 environment.

Css-Style-Name

This property is ignored as the WEB-BROWSER control is not supported in a Web Direct 2.0 environment.

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data for a web-browser

```
procedure division.  
...  
modify screen-1-wb-1 custom-data "Screen-1-custom-data".
```

Enabled

This property assumes a value of "0" if the Web-Browser control is disabled, "1" if it is enabled.

This property is supported only by the JavaFX implementation of the WEB-BROWSER control. It means that it will have effect only if the configuration property [iscobol.gui.webbrowser.class](#) is set to "com.iscobol.fx.JFXWebBrowser".

Example - Define a web-browser control, initially disabled and enable it in procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 28.0 cells  
   enabled 0  
   id 15  
   css-style-name "css-wb"  
   value "http://www.veryant.com"  
   .  
...  
procedure division.  
...  
   modify screen-1-wb-1 enabled 1  
...
```

Event-List

This property specifies a list of events that may or may not be fired depending on the value of the [Exclude-Event-List](#) property. The property requires a sequence of numeric values. It's suggested that you use the constant values defined in the [isgui.def](#) copybook. Multiple values must be enclosed between parenthesis and separated by a space.

Example - Define a web-browser control with a list of excluded events

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 28.0 cells  
   id 15  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   value "http://www.veryant.com"  
   .  
...
```

Exclude-Event-List

If this property is set to "1", then none of the events in the [Event-List](#) property are fired. If this property is set to "0", then only the events listed in the [Event-List](#) property are fired. If it's omitted, then all the events are fired. Preventing the runtime from generating some events may speed up performance in client/server environments.

Note - Excluding focus change events like CMD-GOTO and MSG-VALIDATE may avoid AFTER and BEFORE embedded procedures to be triggered.

Example - Define a web-browser control with a list of excluded events

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 28.0 cells  
   id 15  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   value "http://www.veryant.com"  
   .
```

File-Name

This property specifies the name of the file generated by [Save-As-No-Prompt](#). This property is set to null after each save.

Example - Define a web-browser control and save it as html file in procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 28.0 cells  
   id 15  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   value "http://www.veryant.com"  
   .  
...  
procedure division.  
...  
  modify screen-1-wb-1  
    file-name "c:\main-st\tmp\veryant.html"  
    save-as-no-prompt 1.
```

Font

This property specifies the font that may be used to compute the height and the width of the Web-Browser control. See the [Height-In-Cells](#), [Font](#), [Size](#), and [Width-In-Cells](#) properties for further details.

Example - Define a web-browser control with a font to control the size

```
working-storage section.  
77 Arial-0v0 handle of font.  
...  
screen section.  
...  
    03 screen-1-wb-1 Web-Browser  
        line 23.1  
        column 2.0  
        size 62.2 cells  
        lines 25.6 cells  
        font Arial-0v0  
        id 15  
        value "http://www.veryant.com"  
        .  
...  
procedure division.  
...  
    initialize wfont-data arial-0v0.  
    move 0 to wfont-size.  
    move "Arial" to wfont-name.  
    set wfont-bold to false.  
    set wfont-italic to false.  
    set wfont-underline to false.  
    set wfont-strikeout to false.  
    set wfont-fixed-pitch to false.  
    call "w$font" using wfont-get-font arial-0v0 wfont-data.  
...  

```

Go-Back

When set to a non-zero value, the previous document in the history is loaded.

Example - Define a web-browser control, then in procedure division do a go back

```
screen section.  
...  
    03 screen-1-wb-1 Web-Browser  
        line 23.1  
        column 2.0  
        size 62.2 cells  
        lines 25.6 cells  
        id 15  
        value "http://www.veryant.com"  
        .  
...  
procedure division.  
...  
    modify screen-1-wb-1 go-back 1  
...  

```

Go-Forward

When set to a non-zero value, the next document in the history is loaded.

Example - Define a web-browser control, then in procedure division do a go forward

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
    line 23.1  
    column 2.0  
    size 62.2 cells  
    lines 25.6 cells  
    id 15  
    value "http://www.veryant.com"  
    .  
...  
procedure division.  
...  
    modify screen-1-wb-1 go-forward 1  
...  

```

Go-Home

When set to a non-zero value, the URI set in the `iscobol.gui.web_browser.home` property is loaded. By default, "www.veryant.com" is loaded.

Example - Define a web-browser control, then in procedure division do a go home

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
    line 23.1  
    column 2.0  
    size 62.2 cells  
    lines 25.6 cells  
    id 15  
    value "http://www.veryant.com"  
    .  
...  
procedure division.  
...  
    modify screen-1-wb-1 go-home 1  
...  

```

Go-Search

When set to a non-zero value, the URI set in the `iscobol.gui.web_browser.search` property is loaded. By default, "www.google.com" is loaded.

Example - Define a web-browser control, then in procedure division do a go search

```
screen section.  
...  
  03 screen-1-wb-1 Web-Browser  
    line 23.1  
    column 2.0  
    size 62.2 cells  
    lines 25.6 cells  
    id 15  
    value "http://www.veryant.com"  
    .  
...  
procedure division.  
...  
  modify screen-1-wb-1 go-search 1  
...  

```

Help-Id

This property allows you to assign a unique ID to the Web-Browser control to be passed to the help processor. See [Help automation](#) for more information.

Example - Define a web-browser control with help-id

```
screen section.  
...  
  03 screen-1-wb-1 Web-Browser  
    line 23.1  
    column 2.0  
    size 62.2 cells  
    lines 25.6 cells  
    help-id 2043  
    id 15  
    value "http://www.veryant.com"  
    .  
...  

```

Hint

This property allows you to define the text shown in the window that pops up when the mouse pointer is placed on the Web-Browser control.

Example - Define a web-browser with hint message

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .
```

Id

This property allows you to assign a unique ID to the Web-Browser control.

This is the information the variables *event-control-id* and *control-id* refer to. Both variables are defined in [iscrt.def](#).

Example - Define a web-browser with an ID

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   value "http://www.yahoo.com"  
   .
```

Layout-data

The Layout Manager can use this data to help determine the way to show the control. Each manager forces its own interpretation of the meaning of this data.

This property can have either numeric values (defined in the [isresize.def](#) Copybook) or alphanumeric values, depending on the Layout Manager associated to the window. See [Layout managers](#) for more information.

Example - Define a web-browser with layout data to resize in X and Y if the layout manager requests so, it

also has maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   max-width 120.0  
   min-width 30.0  
   min-height 10.0  
   max-height 50.0  
   layout-data 17  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .
```

Line

This property allows you to specify the Web-Browser control's vertical position. The value is specified in cells. Decimal values are allowed.

When the Web-Browser control is part of a Screen Section, you may specify 'PLUS', '+' or '-' between the property name and its value. The vertical position of the Web-Browser control will be relative to the starting position of the prior Screen Section item.

When the Web-Browser control is part of a Screen Section and the LINE Property is omitted, LINE + 0 is implied.

```
03 Label, LINE 2, LINES 5, (more screen options).  
03 Web-Browser, LINE + 10, (more screen options).
```

The second control will be placed at line 12.

Example - Position a web-browser at line 8.0 on the screen section definition screen section.

```
screen section  
...  
03 screen-1-wb-1 Web-Browser  
   line 8.0  
   column 5.0  
   color 7  
   size 45.0 cells  
   id 2  
   .
```

Lines

This property allows you to specify the height of the Web-Browser control. If the PIXEL keyword follows the value specified here, the height is computed in pixels. If either the CELLS keyword or the [Height-In-Cells](#) style is specified, the height is computed in cells. In this case decimal values are allowed and the cell size is based

on the font used for the parent window.

If the value of this property is not followed by any keyword and the [Height-In-Cells](#) style is not set, the height of the Web-Browser control is still computed in CELLS, but the cell size is based on the font set for the Web-Browser control with the [Font](#) property. If no font has been defined for the Web-Browser control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a web-browser with size in LINES

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 25.6 cells  
   help-id 2043  
   id 15  
   value "http://www.yahoo.com"  
   .
```

Max-Height

The control's maximum height. This setting will affect the Layout Manager's behavior.

Example - Define a web-browser with layout data to resize in X and Y if the layout manager requests so, it also has maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   max-width 120.0  
   min-width 30.0  
   min-height 10.0  
   max-height 50.0  
   layout-data 17  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .
```

Max-Width

The control's maximum width. This setting will affect the Layout Manager's behavior.

Example - Define a web-browser with layout data to resize in X and Y if the layout manager requests so, it

also has maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   max-width 120.0  
   min-width 30.0  
   min-height 10.0  
   max-height 50.0  
   layout-data 17  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .
```

Max-Progress

This property shows the status of end of page loading, i.e. 100% of loading.

Example - Define a web-browser and check the loading progress

```
working-storage section.  
77 ws-progress pic 9(3).  
...  
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   value "http://www.yahoo.com"  
   .  
...  
procedure division.  
...  
   inquire screen-1-wb-1 max-progress ws-progress  
...  
...
```

Min-Height

The control's minimum height. This setting will affect the Layout Manager's behavior.

Example - Define a web-browser with layout data to resize in X and Y if the layout manager requests so, it

also has maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   max-width 120.0  
   min-width 30.0  
   min-height 10.0  
   max-height 50.0  
   layout-data 17  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .
```

Min-Width

The control's minimum width. This setting will affect the Layout Manager's behavior.

Example - Define a web-browser with layout data to resize in X and Y if the layout manager requests so, it also has maximum and minimum dimensions

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   max-width 120.0  
   min-width 30.0  
   min-height 10.0  
   max-height 50.0  
   layout-data 17  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .
```

Navigate-Url

When inquired, this property returns the URL value in navigation field.

Example - Define a web-browser and get the navigate-url value in procedure division

```
working-storage section.  
77 ws-navigate-url pic x(255).  
...  
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   help-id 2043  
   id 15  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .  
  
procedure division.  
...  
   inquire screen-1-wb-1 navigate-url ws-navigate-url  
...  

```

Print

When set to a non-zero value, a dialog is shown allowing the user to choose the active printer and eventually print the content of the page.

Example - Define a web-browser and send the page contents to print on procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   help-id 2043  
   id 15  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .  
  
procedure division.  
...  
   modify screen-1-wb-1 print 1  
...  

```

Print-No-Prompt

When set to a non-zero value, the content of the page is printed on the current printer.

Example - Define a web-browser and send the page contents to print on procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   help-id 2043  
   id 15  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .  
  
procedure division.  
...  
   modify screen-1-wb-1 print-no-prompt 1  
...
```

Progress

This property shows the status of page loading, in terms of percentage.

Example - Define a web-browser and check in procedure division the loading progress percent

```
working-storage section.  
77 ws-progress pic 9(3).  
...  
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   help-id 2043  
   id 15  
   hint "Web Browser - Veryant"  
   value "http://www.yahoo.com"  
   .  
  
procedure division.  
...  
   inquire screen-1-wb-1 progress ws-progress  
...
```

Refresh

When set to a non-zero value, the current URI is reloaded.

Example - Define a web-browser and refresh the loaded page in procedure division

```
screen section.  
...  
  03 screen-1-wb-1 Web-Browser  
    line 23.9  
    column 2.2  
    size 62.2 cells  
    lines 25.6 cells  
    help-id 2043  
    id 15  
    hint "Web Browser - Veryant"  
    value "http://www.yahoo.com"  
    .  
...  
procedure division.  
...  
  modify screen-1-wb-1 refresh 1  
...
```

Save-As

When set to a non-zero value, a Save As dialog is shown allowing the user to save the content of the current page to a file. Only the html code is saved.

Example - Define a web-browser control and save it as html file in procedure division

```
screen section.  
...  
  03 screen-1-wb-1 Web-Browser  
    line 23.1  
    column 2.0  
    size 62.2 cells  
    lines 28.0 cells  
    id 15  
    event-list ( cmd-goto cmd-help )  
    exclude-event-list 1  
    value "http://www.veryant.com"  
    .  
...  
procedure division.  
...  
  modify screen-1-wb-1 save-as 1.
```

Save-As-No-Prompt

When set to a non-zero value, the content of the current page is saved to the file specified by [File-Name](#). Only the html code is saved. If [File-Name](#) is not set, then a Save As dialog is shown.

Example - Define a web-browser control and save it as html file in procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 28.0 cells  
   id 15  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   value "http://www.veryant.com"  
   .  
...  
procedure division.  
...  
  modify screen-1-wb-1  
    file-name "c:\main-st\tmp\veryant.html"  
    save-as-no-prompt 1.
```

Size

This property allows you to specify the size of the Web-Browser control. If the **PIXEL** keyword follows the value specified here, the size is computed in pixels. If either the **CELLS** keyword or the [Width-In-Cells](#) style is specified, the size is computed in CELLS. In this case decimal values are allowed and the cell size is based on the font used for the parent window.

If the value of the property is not followed by any keyword and the [Width-In-Cells](#) style is not set, the size of the Web-Browser control is still computed in CELLS, but the cell size is based on the font set for the Web-Browser control with the [Font](#) property. If no font has been defined for the Web-Browser control, the cell size is based on the font used for the parent window. Decimal values are allowed in this case, too.

Example - Define a web-browser control and set its size

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.1  
   column 2.0  
   size 62.2 cells  
   lines 28.0 cells  
   id 15  
   event-list ( cmd-goto cmd-help )  
   exclude-event-list 1  
   value "http://www.veryant.com"  
   .
```

Status-Text

This property allows you to show the text in the Status bar.

Example - Define a web-browser control with status text

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   hint "Web Browser - Veryant"  
   status-text "Web browser status text"  
   value "http://www.veryant.com"  
   .
```

Stop-Browser

When set to a non-zero value, the Web-Browser control stops loading the URI.

Example - Define a web-browser control and stop the page loading in procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   status-text "Web browser status text"  
   value "http://www.veryant.com"  
   .  
...  
procedure division.  
...  
   modify screen-1-wb-1 stop-browser 1  
...  
...
```

Title

This property allows you to retrieve the title of the currently loaded URI.

Example - Define a web-browser control and inquire the title in procedure division

```
working-storage section.  
77 ws-title    pic x(255).  
...  
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   status-text "Web browser status text"  
   value "http://www.veryant.com"  
   .  
...  
procedure division.  
...  
   inquire screen-1-wb-1 title ws-title  
...  

```

Value

This property represents the value of the Web-Browser control.

When inquired, it returns the value that is currently represented.

When set, the Web-Browser control changes its look to represent it.

It is the current URI.

Example - Define a web-browser control and set the URL value

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   value "http://www.veryant.com"  
   .  

```

Visible

This property assumes a value of "0" if the Web-Browser control is not visible, "1" if it is visible.

Example - Define a web-browser control initially invisible and set it visible in procedure division

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   font Arial-0v0  
   help-id 2043  
   id 15  
   value "http://www.veryant.com"  
   visible 0  
   .  
...  
procedure division.  
...  
   modify screen-1-wb-1 visible 1  
...
```

Styles

The following styles are applicable to the WEB-BROWSER control: [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bold](#), [Height-In-Cells](#), [High](#), [Highlight](#), [Low](#), [Lowlight](#), [No-Msg-Before-Navigate](#), [Permanent](#), [Standard](#), [Temporary](#), [Use-Alt](#), [Use-Return](#), [Use-Tab](#), [Width-In-Cells](#).

{ [Background-High](#) | [Background-Low](#) | [Background-Standard](#) }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a web-browser control with high background and bold foreground

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   background-high bold  
   value "http://www.veryant.com"  
   .
```

Height-In-Cells

This style implies that the value specified for the [Lines](#) property, the one setting the Web-Browser control's height, is expressed in CELLS. It is possible to get the same result writing: "[Lines](#) *value* CELLS".

Example - Define a web-browser control with height and width in cells using the styles

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2  
   lines 25.6  
   id 15  
   height-in-cells  
   width-in-cells  
   value "http://www.veryant.com"  
   .
```

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Define a web-browser control with high background and bold foreground

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   background-high bold  
   value "http://www.veryant.com"  
   .
```

No-Msg-Before-Navigate

This style prevents the generation of the MSG-BEFORE-NAVIGATE event. This is useful when

- the web site that you're navigating must send data thru POST (e.g. a login form) and the MSG-BEFORE-NAVIGATE event handling performed by isCOBOL causes parameters to get lost.
- the Web-Browser is used to browse disk folders, by setting the **Value** property to a URL that begins with "file:///".
- the Web-Browser is used to show a file with an embedded component, for example if you load a PDF file that is rendered through the Acrobat ActiveX embedded in the browser.

This style is automatically applied if `iscobol.gui.webbrowser.no_msg_before_navigate` (boolean) is set to true in the configuration.

Example - Define a web-browser control with no-msg-before-navigate

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   no-msg-before-navigate  
   value "http://www.veryant.com"  
   .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Define a web-browser control with temporary style

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   temporary  
   value "http://www.veryant.com"  
   .
```

Use-Alt

If this style is set, the Alt key is not trapped by the control and can be intercepted by the COBOL program.

Example - Define a web-browser that allows to intercept the ALT, RETURN and TAB keys

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   use-alt  
   use-return  
   use-tab  
   value "http://www.veryant.com"  
   .
```

Use-Return

If this style is set, the Enter key is not trapped by the control and can be intercepted by the COBOL program.

Example - Define a web-browser that allows to intercept the ALT, RETURN and TAB keys

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   use-alt  
   use-return  
   use-tab  
   value "http://www.veryant.com"  
   .
```

Use-Tab

If this style is set, the Tab key is not trapped by the control and can be intercepted by the COBOL program.

Example - Define a web-browser that allows to intercept the ALT, RETURN and TAB keys

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2 cells  
   lines 25.6 cells  
   id 15  
   use-alt  
   use-return  
   use-tab  
   value "http://www.veryant.com"  
   .
```

Width-In-Cells

This style implies that the value specified for the [Size](#) property, the one setting the Web-Browser control's width, is expressed in CELLS. It is possible to get the same result writing: "[Size value CELLS](#)".

Example - Define a web-browser control with height and width in cells using the styles

```
screen section.  
...  
03 screen-1-wb-1 Web-Browser  
   line 23.9  
   column 2.2  
   size 62.2  
   lines 25.6  
   id 15  
   height-in-cells  
   width-in-cells  
   value "http://www.veryant.com"  
   .
```

Events

The following events are applicable to the WEB-BROWSER control: [MSG-WB-BEFORE-NAVIGATE](#), [MSG-WB-DOWNLOAD-BEGIN](#), [MSG-WB-DOWNLOAD-COMPLETE](#), [MSG-WB-NAVIGATE-COMPLETE](#), [MSG-WB-PROGRESS-CHANGE](#), [MSG-WB-STATUS-TEXT-CHANGE](#), [MSG-WB-TITLE-CHANGE](#).

MSG-WB-BEFORE-NAVIGATE

This event is fired before a new URI is loaded. Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the Web-Browser control from loading the URI.

MSG-WB-DOWNLOAD-BEGIN

This event is fired when the Web-Browser control starts loading a new URI.

MSG-WB-DOWNLOAD-COMPLETE

This event is fired when the Web-Browser control has finished loading a new URI.

MSG-WB-NAVIGATE-COMPLETE

This event is fired when a new URI has been completely loaded. The [Value](#) property is set to the value of the new URI.

MSG-WB-PROGRESS-CHANGE

This event is fired each time a download proceeds. The [Progress](#) property is updated and can be inquired to retrieve what part of the document has been downloaded.

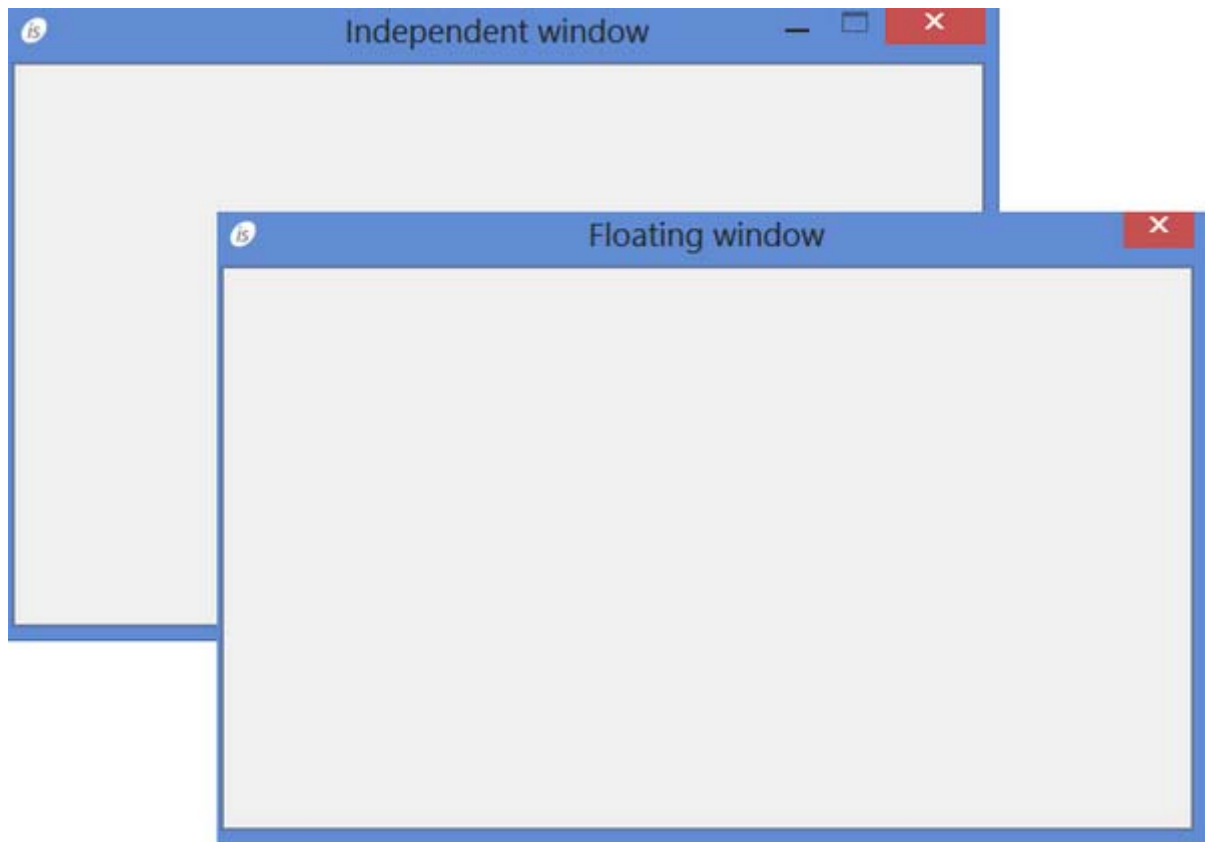
MSG-WB-STATUS-TEXT-CHANGE

This event is fired when the text in the Web-Browser control's status-bar changes. The [Status-Text](#) property is updated.

MSG-WB-TITLE-CHANGE

This event is fired when the title of the current document changes.

WINDOW



Windows are containers of controls and can be resized or moved around.

Windows are created with the [DISPLAY](#) Statement, Format 10.

There are four kinds of window:

- **INITIAL (or STANDARD):** The main window of the COBOL application. There can be only one and therefore it's usually used to store the application's main screen. It appears as a button in the task bar. If you minimize it, all other windows of the same application are minimized. If you close it, the application terminates.
- **INDEPENDENT:** The most common type of window, typically used to store the screen of the COBOL application programs. It appears as a button in the task bar. If you click on the close button, a **CMD-CLOSE** event is generated only if the **System Menu** style is set, otherwise nothing happens.
- **FLOATING:** A pop-up panel, bound to its parent window (which can be INDEPENDENT or INITIAL). By default, this kind of window is modal, which means the user cannot switch back to the parent window until the FLOATING window is closed. It does not appear as a button in the task bar and cannot be minimized, maximized or restored. If you click on the close button, a **CMD-CLOSE** event is only generated if the **System Menu** style is set, otherwise nothing happens.
- **DOCKING:** A container of DOCKABLE windows. It has the same characteristics as the INDEPENDENT window, but instead of storing a standard screen, it stores multiple dockable windows, each one with its own screen.
- **MDI-PARENT:** A container of MDI-CHILD windows. It has the same characteristics as the INDEPENDENT window, but instead of storing a standard screen, it stores other windows, each one with its own screen. The child windows can be moved by the user inside the MDI-PARENT window area, but they never get outside of it.

Properties

The following properties are applicable to the WINDOW control: [Action](#), [Background-Color](#), [Cell Height](#), [Cell Size](#), [Cell Width](#), [Col](#), [Color](#), [Column](#), [Control Font](#), [Custom-Data](#), [Enabled](#), [Font](#), [Foreground-Color](#), [Gradient-Color-1](#), [Gradient-Color-2](#), [Gradient-Orientation](#), [Help-Id](#), [Hint](#), [Icon](#), [Layout](#) [Layout-Manager](#), [Line](#), [Lines](#), [Max-Lines](#), [Max-Size](#), [Min-Lines](#), [Min-Size](#), [Pop-Up Menu](#), [Pos](#), [Position](#), [Screen Col](#), [Screen Column](#), [Screen Line](#), [Screen Pos](#), [Screen Position](#), [Screen-Index](#), [Size](#), [Title](#), [Visible](#).

Action

A specific action is performed when a value is assigned to this property. The following symbolic values, included in the copy file [isgui.def](#), can be assigned. The table below shows the actions related to each value:

action-maximize	The window is maximized.
action-minimize	The window is minimized.
action-restore	The window is restored, after being minimized.

Example - Display a standard window and maximize it later with the action property

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...  
    modify window-handle action action-maximize  
...
```

Background-Color

This property allows you to set or retrieve the background color of the Window control.

The value set here is also the default background color of the controls created in the Window control.

See "[Color management](#)" for further details.

Example - Display a standard window with background and foreground colors

```
77 window-handle usage handle of window.
...
procedure division.
...
    display standard window background-low
        screen line 41
        screen column 91
        size 64.0
        lines 55.8
        cell width 10
        cell height 10
        label-offset 20
        control font Default-Font
        background-color 15
        foreground-color 1
        resizable
        modeless
        title-bar
        no wrap
        erase
        title "Screen"
        handle window-handle
    .
...
```

Cell Height

This property defines the height of the cell used for positioning the various controls inside the Window control.

The value must comply with the following specification:

```
{ Pixels }
{ ControlType FONT [FontHandle] [SEPARATE] }
{ [OVERLAPPED] }
```

Refer to the [Cell Size](#) property for a more detailed explanation.

Example - Display a standard window with cell height and width

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        background-color 15  
        foreground-color 1  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .
```

Cell Size

This property defines the size (both height and width) of the cell used for positioning the various controls inside the Window control.

The value must comply with the following specification:

```
{ ControlType FONT [FontHandle] [SEPARATE]  }  
{                                           [OVERLAPPED] }
```

The cell size is the space the ControlType needs to show the character "0" without truncating it. ControlType can be LABEL, the default, or ENTRY-FIELD. When ENTRY-FIELD is specified, the cell is larger.

FontHandle is the font used to calculate the cell size. If not specified, the font set in the [Control Font](#) property will be used.

When the SEPARATE clause is specified, the cell size is increased, so that controls placed on contiguous lines or columns are not too close to each other.

When the OVERLAPPED clause is specified, the cell size is left unchanged and controls placed on contiguous lines or columns look joint. This is the default setting.

Example - Display a standard window with cell size in one property only

```
working-storage section.  
77 window-handle usage handle of window.  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        label-offset 20  
        control font Default-Font  
        background-color 15  
        foreground-color 1  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        cell size 10  
        handle window-handle  
    .
```

Cell Width

This property defines the width of the cell used for positioning the various controls inside the Window control.

The value must comply with the following specification:

```
{ Pixels }  
{ ControlType FONT [FontHandle] [SEPARATE] }  
{ [OVERLAPPED] }
```

Refer to the [Cell Size](#) property for a more detailed explanation.

Example - Display a standard window with cell height and width

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        background-color 15  
        foreground-color 1  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .
```

[Col | Column | Pos | Position]

This property allows you to specify the Window control's horizontal position. The value is specified in cells. Decimal values are allowed. The position is relative to the parent window.

Example - Display a new window at line 2, column 3

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display independent window background-low  
        line 2.0  
        column 3.0  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        resizable  
        modeless  
        title-bar  
        title "Screen"  
        handle window-handle  
    .
```

Color

This property allows you to set or retrieve the color of the Window control.

The value set here is also the default color of the controls created in the Window control.

Foreground and background color values are combined and therefore RGB colors are not supported. See ["Color management"](#) for further details.

Example - Display a standard window using the color property to set the foreground and background colors

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 260  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
        .
```

Control Font

Since the Window control is a container, the controls it contains may inherit some peculiarities.

The font handle set to this property will be the default value of the FONT Property of the controls created in the Window control.

This property also affects the way the cell size is calculated. See the [Cell Height](#), [Cell Size](#), and [Cell Width](#) properties for further details.

Example - Display a standard window using the control font property

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 260  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .
```

Custom-Data

This property allows you to set or retrieve the hidden value of the control. The hidden value is never shown to the user, its purpose is to give the programmer an easy way to store and retrieve information related to the control.

Example - Set the custom data of a window

```
procedure division.  
...  
    modify window-handle custom-data "Screen 1 custom data"  
...  
.
```

Enabled

This property assumes a value of "0" if the Window control is disabled, "1" if it is enabled.

Example - Display a standard window and disable it later

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 260  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...  
    modify window-handle enabled 0  
...
```

Font

The font handle set to this property will be used for character-based DISPLAY Statements.

Example - Display a standard window and use the font property

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        font Default-Font  
        color 260  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .
```

Foreground-Color

This property allows you to set or retrieve the foreground color of the Window control.

The value set here is also the default foreground color of the controls created in the Window control.

See "[Color management](#)" for further details.

Example - Display a standard window with background and foreground colors

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        background-color 15  
        foreground-color 1  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...
```

Gradient-Color-1

This property allows you to set or retrieve the start color of the gradient effect of the Window control.

If this property is not set, but [Gradient-Color-2](#) is set, then the start color of the gradient effect is black.

The gradient effect overrides the color set by [Background-Color](#).

See "[Color management](#)" for further details.

Example - Display a standard window whose background color goes from gray to white

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        gradient-color-1 rgb x#c0c0c0  
        gradient-color-2 rgb x#ffffff  
        gradient-orientation 0  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...
```

Gradient-Color-2

This property allows you to set or retrieve the end color of the gradient effect of the Window control.

If this property is not set, but [Gradient-Color-1](#) is set, then the end color of the gradient effect is black.

The gradient effect overrides the color set by [Background-Color](#).

See "[Color management](#)" for further details.

Example - Display a standard window whose background color goes from gray to white

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        gradient-color-1 rgb x#c0c0c0  
        gradient-color-2 rgb x#ffffff  
        gradient-orientation 0  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...
```

Gradient-Orientation

This property allows you to set or retrieve the orientation of the gradient effect of the Window control.

Possible values are:

Value	Orientation
0	North to South
1	Northeast to Southwest
2	East to West
3	Southeast to Northwest
4	South to North
5	Southwest to Northeast
6	West to East
7	Northwest to Southeast

Constants for the above values are provided in the [isgui.def](#) copybook.

If this property is not set, the default orientation is North to South.

Example - Display a standard window whose background color goes from gray to white

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        gradient-color-1 rgb x#c0c0c0  
        gradient-color-2 rgb x#ffffff  
        gradient-orientation 0  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
        .  
...
```

Help-Id

This property is ignored by windows.

Hint

This property is ignored by windows.

Icon

When this property is set to a bitmap handle, the corresponding image becomes the window icon.

Custom icons can be used on INITIAL/STANDARD windows.

By setting the [iscobol.gui.independent.icon \(boolean\)](#) configuration property to true, custom icons can be used also on INDEPENDENT, DOCKING and MDI-PARENT windows.

Custom icons can't be used on FLOATING, DOCKABLE and MDI-CHILD windows, instead.

Example - Display a window with a different icon

```
working-storage section.  
77 window-handle usage handle of window.  
77 icon-ico      pic s9(9) comp-4.  
...  
procedure division.  
...  
    call "w$bitmap"  
        using wbitmap-load "icon.png"  
        giving icon-ico.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        icon icon-ico  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 260  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        hint "App main window"  
        handle window-handle  
    .
```

Layout

This property applies only to docking windows and allows you to set and retrieve the layout of a docking window.

The isCOBOL docking window layout manager recursively arranges its components in row and column groups called Splits. Elements of the layout are separated by gaps called Dividers. The overall layout is defined with a simple tree model. Named Leaf nodes represent the space allocated to a component that was added with a constraint that matches the Leaf's name. Extra space is distributed among row and column siblings according to their weight. The weight value varies from 0.0 to 1.0. If no weights are specified then the last sibling always gets all of the extra space, or space reduction.

Nodes are represented by parenthetical expressions whose first token is one of ROW/COLUMN/LEAF. ROW and COLUMN specify horizontal and vertical Split nodes respectively, LEAF specifies a Leaf node. A Leaf's name and weight can be specified with attributes, name=myLeafName weight=myLeafWeight. Similarly, a Split's weight can be specified with weight=value. For example, the following expression generates a horizontal Split node with three children: the Leafs named left and right, and a Divider in between:

```
(ROW (LEAF name=left) (LEAF name=right weight=1.0))
```

Dividers must not be included in the string, they're added automatically as needed. Because Leaf nodes often only need to specify a name, one can specify a Leaf by just providing the name.

The previous example can be written like this:

```
(ROW left (LEAF name=right weight=1.0))
```

Here's a more complex example. One row with three elements, the first and last of which are columns with two leaves each:

```
(ROW (COLUMN weight=0.5 left.top left.bottom)
      (LEAF name=middle)
      (COLUMN weight=0.5 right.top right.bottom))
```

It's also possible to specify the minimum number of pixels that the dockable window must retain when it's resized inside of the docking window with the optional attribute minpixel. The minpixel attribute acts on the height or on the width depending on the kind of node where it appears. In the example below, the dockable window on the right cannot be horizontally resized under 30 pixels:

```
(ROW left (LEAF name=right weight=1.0 minpixel=30))
```

Example - Display a docking window with a layout

```
working-storage section.
77 var-layout          pic x(200).
77 var-leaf1           pic x(10).
77 var-leaf2           pic x(10).
77 h-docking           handle of window.
...
procedure division.
...
    move "(ROW (COLUMN (LEAF name=tl weight=0.5)"&
          "(LEAF name=bl weight=0.5) weight=0.5)"&
          "(COLUMN (LEAF name=tr weight=0.5)"&
          "(LEAF name=br weight=0.5) weight=0.5))"
          to var-layout.

    display docking window
           resizable
           lines      24
           size       75
           min-lines  24
           min-size   75
           title      "Docking Window"
           layout     var-layout
           handle     h-docking.
```

Layout-Manager

This property associates a Layout Manager with the Window. Three types of Layout Managers (defined in the

isresize.def Copybook) are supported.

- LM-RESIZE
- LM-RESPONSIVE
- LM-SCALE

Example - Display a window with a layout manager

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 64.0  
        lines 55.8  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 260  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        layout-manager lm-resize  
        handle window-handle  
    .
```

Line

This property allows you to specify the Window control's vertical position. The value is specified in cells. Decimal values are allowed. The position is relative to the parent window.

Example - Display a window with a line and column relative to the main window

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

Lines

This property allows you to specify the height of the Window control. Decimal values are allowed.

Example - Display a window with height in lines

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

Max-Lines

The value set here establishes the maximum height of the window. This property is used in conjunction with either the [RESIZABLE](#) style or the [AUTO-RESIZE](#) style and has no effect on docking windows.

Example - Display a window with resizable style and maximum and minimum dimensions when resizing

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        handle window-handle  
    .
```

Max-Size

The value set here establishes the maximum width of the window. This property is used in conjunction with either the [RESIZABLE](#) style or the [AUTO-RESIZE](#) style and has no effect on docking windows.

Example - Display a window with resizable style and maximum and minimum dimensions when resizing

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        handle window-handle  
    .
```

Min-Lines

The value set here establishes the minimum height of the window. This property is used in conjunction with either the [RESIZABLE](#) style or the [AUTO-RESIZE](#) style and has no effect on docking windows.

Example - Display a window with resizable style and maximum and minimum dimensions when resizing

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        handle window-handle  
    .
```

Min-Size

The value set here establishes the minimum width of the window. This property is used in conjunction with either the [RESIZABLE](#) style or the [AUTO-RESIZE](#) style and has no effect on docking windows.

Example - Display a window with resizable style and maximum and minimum dimensions when resizing

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        handle window-handle  
    .
```

Pop-Up Menu

With this property it is possible to associate a pop-up menu with the Window control by assigning a pop-up menu handle to it. The [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#) and [MSG-END-MENU](#) events may be generated.

Example - Display a window with pop-up menu

```
working-storage section.
77 window-handle usage handle of window.
77 hmenu pic s9(9) comp-4.
...
procedure division.
...
*> Use w$menu to build the pop-up menu prior to display the window
...
    display standard window background-low
        screen line 41
        screen column 91
        size 64.0
        lines 48.0
        cell width 10
        cell height 10
        label-offset 20
        control font Default-Font
        color 257
        modeless
        title-bar
        no wrap
        erase
        title "Screen"
        handle window-handle
        pop-up menu hmenu
    .
...

```

[Screen Col | Screen Column | Screen Pos | Screen Position]

This property allows you to set or retrieve the horizontal position of the window in the screen. It is measured in pixels and the origin of the screen is at 0.

Example - Display a window on a specific screen column and screen line

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        handle window-handle  
    .  
...
```

Screen Line

This property allows you to set or retrieve the vertical position of the window in the screen. It is measured in pixels and the origin of the screen is at 0.

Example - Display a window on a specific screen column and screen line

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...  
.
```

Screen-Index

This property allows to set or retrieve the ordinal number of the monitor where the window is positioned in a multi-monitor environment.

Call the [C\\$MONITOR](#) routine to know how many monitors are available.

The default value is 0. In this case the positioning on a different monitor is controlled by the window coordinates specified by SCREEN LINE and SCREEN COL: if the coordinates have negative values, then the window is displayed in the monitor left to the current one. If the coordinates have higher values, then the window is displayed in the monitor right to the current one. If SCREEN-INDEX is set to a value greater than zero instead, then SCREEN LINE and SCREEN COL are relative to the monitor pointed by SCREEN-INDEX.

Example - Display a window on monitor number 1, then move it to monitor number 2:

```
working-storage section.
77 window-handle usage handle of window.
77 hmenu pic s9(9) comp-4.
...
procedure division.
...
*> Use w$menu to build the pop-up menu prior to display the window
...
        display standard window background-low
            screen-index 1
            size 64.0
            lines 48.0
            cell width 10
            cell height 10
            label-offset 20
            control font Default-Font
            color 257
            modeless
            title-bar
            no wrap
            erase
            title "Screen"
            handle window-handle
            pop-up menu hmenu
        .
...
        modify window-handle screen-index 2.
...
```

Size

This property allows you to specify the width of the Window control. Decimal values are allowed.

Example - Display a window with a specific size

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle  
    .  
...  

```

Title

It is the description shown in the Window title bar. When set, the [Title-Bar](#) style is automatically set.

Example - Display a window with a specific title

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Main Menu"  
        handle window-handle  
    .  
...  

```

Visible

This property assumes a value of "0" if the Window control is not visible, "1" if it is visible.

Example - Display a window and make it invisible later

```
working-storage section.
77 window-handle usage handle of window.
...
procedure division.
...
    display standard window background-low
        screen line 41
        screen column 91
        size 22.1
        lines 18.1
        cell width 10
        cell height 10
        label-offset 20
        control font Default-Font
        color 257
        resizable
        modeless
        title-bar
        no wrap
        erase
        title "Main Menu"
        handle window-handle
    .
...
    modify window-handle visible 0
...

```

Styles

The following styles are applicable to the WINDOW control: [Auto-Resize](#), [Background-High](#), [Background-Low](#), [Background-Standard](#), [Bind To Thread](#), [Blank](#), [Bold](#), [Boxed](#), [Controls-Uncropped](#), [High](#), [Highlight](#), [Link To Thread](#), [Low](#), [Lowlight](#), [Modal](#), [Modeless](#), [No Scroll](#), [No Wrap](#), [No-Close](#), [Permanent](#), [Resizable](#), [Reverse](#), [Shadow](#), [Standard](#), [System Menu](#), [Temporary](#), [Title-Bar](#), [Undecorated](#), [User-Colors](#), [User-Gray](#), [User-White](#).

{ Auto-Resize | Resizable }

When either the AUTO-RESIZE style or the RESIZABLE style is set, the window can be resized by the user. Minimum and maximum size are defined by the [Min-Size](#), [Min-Lines](#), [Max-Size](#) and [Max-Lines](#) properties.

Auto-Resize	The window can be resized and scroll-bars are automatically provided if the window becomes smaller than its content. No event is fired.
Resizable	The window can be resized and the NTF-RESIZED event is fired. Recalculating the content position and size is committed to the programmer.

For windows without the RESIZABLE style, [Min-Size](#), [Min-Lines](#), [Max-Size](#) and [Max-Lines](#) are ignored.

Example - Display a window with resizable style and maximum and minimum dimensions when resizing

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        handle window-handle  
    .  
...
```

{ Background-High | Background-Low | Background-Standard }

Background-High	The background color is forced to be bright.
Background-Low	The background color is forced not to be bright.
Background-Standard	The background color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Display a window with background-low and foreground low

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        low  
        handle window-handle  
    .  
...
```

Blank

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

Boxed

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

Controls-Uncropped

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

{ [**Bold** | **High** | **Highlight**] | [**Low** | **Lowlight**] | **Standard** }

Bold, High, Highlight	The foreground color is forced to be bright.
Low, Lowlight	The foreground color is forced not to be bright.
Standard	The foreground color is left unchanged, the default.

Setting this style with RGB colors has no effect. See "[Color management](#)" for further details.

Example - Display a window with background-low and foreground low

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        low  
        handle window-handle  
    .  
...
```

[Link To Thread | Bind To Thread]

Setting any of these styles has no effect. The compiler only accepts them for compatibility reasons.

{ Modal | Modeless }

Modal	The user cannot activate a different window of the same application.
Modeless	The user can click another window of the same application to activate it.

Example - Display a window with modeless style

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

No Scroll

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

No Wrap

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

No-Close

When this style is set, the window close facility is disabled.

Example - Display a window with no-close style

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        no-close  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

{ Permanent | Temporary }

Permanent	A control with this style applied can be destroyed only by using the DESTROY statement or when the parent window is destroyed. A permanent control is more efficient than a "temporary" one since it is not created each time a DISPLAY statement is executed. This style is set by default.
Temporary	Temporary controls are destroyed when a DESTROY statement is executed, when the parent window is destroyed, when another control is created in its same position, or when a Screen Section containing a BLANK SCREEN keyword is displayed. Temporary controls are less efficient than "permanent" ones since they are created each time a DISPLAY statement is executed.

Example - Display a window with temporary style

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        temporary  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

Reverse

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

Shadow

Setting this style has no effect. The compiler only accepts it for compatibility reasons.

System Menu

When this style is set, the system menu becomes available. It usually gives access to commands such as minimize, maximize and close.

Example - Display a window with system menu

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        no-close  
        no wrap  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

Title-Bar

When this style is set, an empty title bar is shown. Use the [Title](#) property to set the title.

Example - Display a window with title-bar style

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display standard window background-low  
        screen line 41  
        screen column 91  
        size 22.1  
        lines 18.1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        resizable  
        modeless  
        title-bar  
        no wrap  
        erase  
        title "Screen"  
        max-size 50.0  
        min-size 5.0  
        min-lines 5.0  
        max-lines 30.0  
        low  
        handle window-handle  
    .
```

Undecorated

When this style is set, native decorations like frame and title bar are not shown.

This style is suggested when using LAFs that provide their own frame and title bar. Without the Undecorated style, these LAFs show a double title-bar.

Example - Display a window with undecorated style

```
working-storage section.  
77 window-handle usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        no-close  
        no wrap  
        erase  
        undecorated  
        title "Screen"  
        handle window-handle1  
    .
```

{ User-Colors | [User-Gray | User-White] }

User-Colors	Combines the effect of the USER-GRAY and USER-WHITE styles.
User-Gray	Controls with the attribute 8 set for the background or the foreground are not actually painted with RGB 192, 192, 192, but with the color defined for the system 3-D objects.
User-White	Controls with the attribute 16 set for the background or the foreground are not actually painted with RGB 255, 255, 255, but with the color defined for the system window background.

Example - Display a window with User-gray style

```
working-storage section.  
77 window-handle1 usage handle of window.  
...  
procedure division.  
...  
    display floating window background-low  
        size 37.3  
        lines 34.2  
        line 20  
        column 1  
        cell width 10  
        cell height 10  
        label-offset 20  
        control font Default-Font  
        color 257  
        modeless  
        system menu  
        title-bar  
        user-gray  
        erase  
        title "Screen"  
        handle window-handle1  
    .
```

Events

The following events are applicable to the WINDOW control: [CMD-ACTIVATE](#), [CMD-CLOSE](#), [MSG-DEICONIFIED](#), [MSG-CLOSE](#), [MSG-END-MENU](#), [MSG-ICONIFIED](#), [MSG-INIT-MENU](#), [MSG-MENU-INPUT](#), [NTF-RESIZED](#).

CMD-ACTIVATE

This event is fired when the user activates a window belonging to the program that is currently running.

CMD-CLOSE

This event is fired when the user has pressed the close icon in the top-right corner of an independent or floating window. The window is not automatically destroyed: this action has to be performed by the program. The EVENT-WINDOW-HANDLE data item contains the handle of the window that has generated the event.

MSG-CLOSE

This event is fired when the user has pressed the close icon in the top-right corner of a standard window. The window is automatically destroyed unless the program assigns the symbolic value "EVENT-ACTION-FAIL" to the EVENT-ACTION data item. The EVENT-WINDOW-HANDLE data item contains the handle of the window that has generated the event.

MSG-DEICONIFIED

This event is fired when the user has pressed the restore icon in the top-right corner of a window. The window is automatically restored from the system tray.

MSG-END-MENU

This event is fired when a menu is removed from the screen. This happens when the user selects a menu item, after the [MSG-MENU-INPUT](#) event, or when the user closes the menu without selecting it. The program should undo some actions here that occurred in the [MSG-INIT-MENU](#) event.

MSG-ICONIFIED

This event is fired when the user has pressed the minimize icon in the top-right corner of a window. The window is automatically minimized to the system tray.

MSG-INIT-MENU

This event is fired right before displaying a menu. The EVENT-DATA-2 data item contains the menu handle and can be used to alter the menu items.

Setting EVENT-ACTION to EVENT-ACTION-FAIL prevents the menu from being displayed.

MSG-MENU-INPUT

This event is fired when the user selects a menu item. The EVENT-DATA-2 data item contains the ID of the menu item that has been selected.

Setting EVENT-ACTION to EVENT-ACTION-CONTINUE prevents the menu from generating an Exception value. This is used when the programmer wants to handle menu actions in the Event Procedure.

NTF-RESIZED

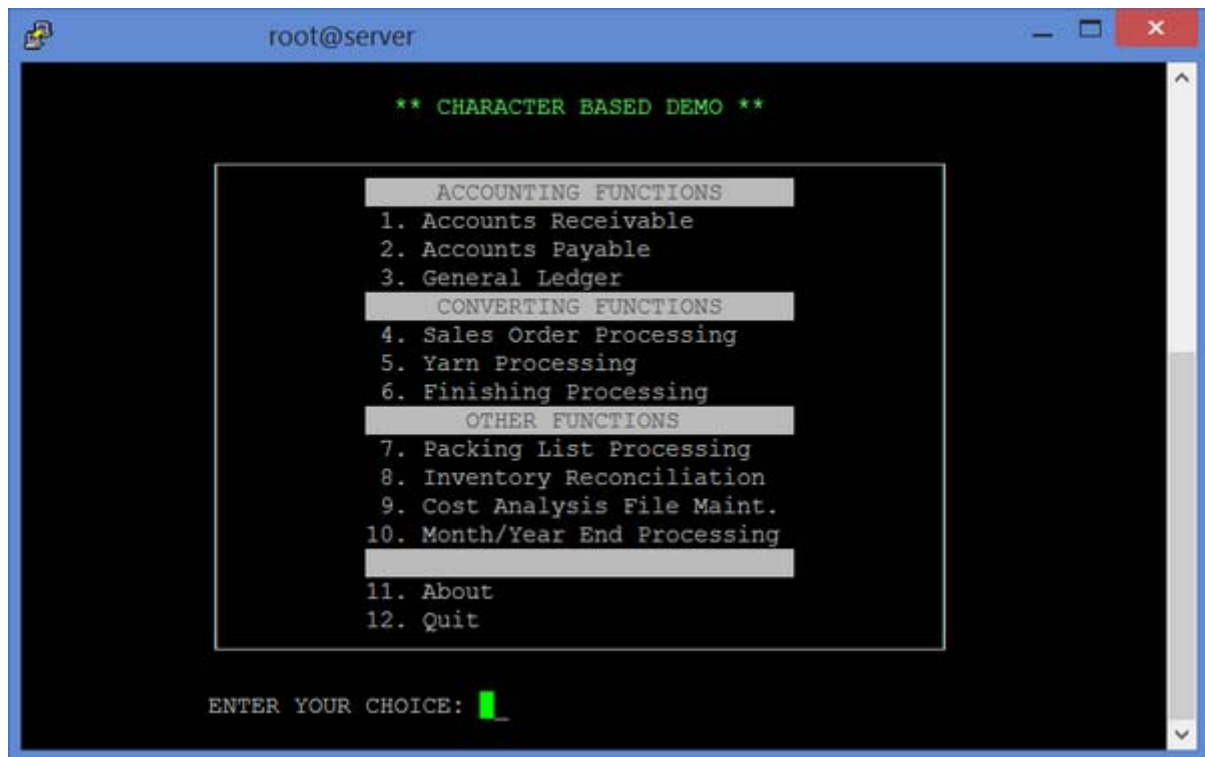
This event is fired when the user tries to resize a window created with the [RESIZABLE](#) style. EVENT-DATA-1 and EVENT-DATA-2 contain the new height and the new width of the window respectively expressed in hundreds of cell.

The user can resize a window by left clicking on the border and dragging the mouse. Depending on your operating system configuration you might obtain only one event when the user releases the mouse button or multiple events while the user is dragging the mouse. In the second case it's good practice to set [iscobol.gui.ntf_resized_delay](#) in the configuration in order to reduce the number of events raised and improve performance.

Chapter 3

Character Based Screens

isCOBOL can run applications with character based screens (green screens, text mode screens) on hardware terminals and terminal emulators.



By default, isCOBOL emulates character based screens using graphical resources. This behavior produces an error if working on terminals that don't include a graphical interface (dumb terminals and terminal emulators). The error returned is:

```
No X11 DISPLAY variable was set, but this program performed an operation which requires it.
```

To avoid this error and enable an application to use the terminal as the screen, isCOBOL takes advantage of CHARVA, a Java Windowing Toolkit for Text Terminals. See [Using CHARVA](#) for more information.

Note that the CHARVA solution is suitable only for those COBOL application whose UI is fully character-based. Graphical UI is not supported by CHARVA.

CHARVA is not supported on the Windows 64 bit platform.

Using CHARVA

isCOBOL supports any terminal that has a description in the terminfo database on the system; in other words, all popular terminals such as VT100, VT220, Wyse and ANSI terminals and the "xterm" and "PuTTY" terminal emulators are supported. Support for UTF-8 character sets is also included (e.g. for Hungarian, Czech, Cyrillic, Korean etc). This feature is implemented using the CHARVA Toolkit.

CHARVA was designed to bring the power and flexibility of Java to applications on Linux/Unix systems (and has also been ported to MS Windows).

Terminal-based applications can now benefit from Java features such as object orientation, multithreading, automatic garbage-collection, and a vast range of libraries.

How to run the program

CHARVA has been embedded in the isCOBOL Framework. In order to enable its use, the following entry must be set in the configuration:

```
iscobol.guifactory.class=com.iscobol.gui.client.charva.GuiFactoryImpl
```

The above setting is automatically activated by the -t option provided by iscrun, so in order to run a program with CHARVA, it's enough to issue the command:

```
iscrun -t PROGRAM_NAME
```

In order to make isCOBOL work correctly with CHARVA, the terminal library (Terminal.dll on Windows and libterminal.so on Unix) must be available in the Java library path while charva.jar, commons-logging.jar and commons-logging-api.jar must be listed in the CLASSPATH.

CHARVA is for purely character-based applications and should not be used with graphical controls. Displaying graphical controls does not usually have any adverse effects, but it can cause unpredictable behavior.

Colors are not enabled by default. In order to activate support for colors, the charva.colors Java property must be set to True. For example:

```
iscrun -t -J-Dcharva.color=1 PROGRAM_NAME
```

Note: In isCOBOL, the background and foreground colors are mapped in the RGB color model, but the CHARVA toolkit does not recognize the attribute highlight in the RGB color value so, when either blue or green or red is greater than 192, the bold attribute is set for the control.

How to debug

Programs that run with CHARVA cannot be debugged directly since the Debugger traps the display that should be redirected to the console, causing Exception errors.

In order to debug a program that runs with CHARVA, the Remote Debugger must be used. So, for example:

```
iscrun -t -J-Dcharva.color=1 -J-Discobol.rundebbug=2 PROGRAM_NAME  
iscrun -d -r
```

Known limitations and differences between Charva and emulated character mode

DISPLAY MESSAGE BOX is not supported. Message box text can be printed on the console if `iscobol.display_message` configuration property is set to 1 or 2.

When using CHARVA, `iscobol.terminal.cursor_type` is ignored: the cursor has the default shape provided by the current terminal. In addition, `iscobol.terminal.autowrap` (boolean) is ignored: the display always wraps.

isCOBOL maps the background and foreground colors to the RGB color model, but CHARVA does not recognize the *highlight* attribute in the RGB color value, therefore isCOBOL applies the *highlight* attribute to the control when either the blue or green or red value is greater than 128.

The `WFONT-GET-FONT` function works only if `WFDEVICE-WIN-PRINTER` is set to TRUE, otherwise `WFONERR-FONT-NOT-FOUND` is returned.

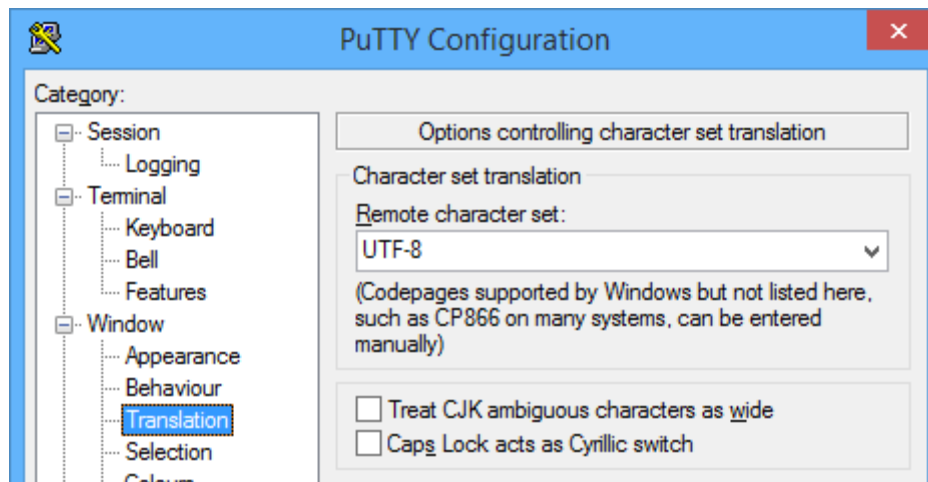
If the program performs only DISPLAY statements without accepting user input with a ACCEPT statement, you need to call the `WFLUSH-REFRESH` at the end in order to make the displayed text appear on video.

PuTTY configuration

When PuTTY is used to connect to a Unix server and run a COBOL program through CHARVA, you should set the *Character set translation* to match the locale of the server. The locale of the server can be retrieved by checking the LANG environment variable:

```
echo $LANG
```

For example, if the locale of the server is "UTF-8", configure PuTTY as follows:



When working with UTF-8, the following setting should also be set:

```
export NCURSES_NO_UTF8_ACS=1
```

The nonobservance of the above suggestions may lead to bad display of grave letters and drawings.

Keyboard shortcuts

To force a refresh of the screen, press CTRL-L.

During the Accept, not all the keyboard shortcuts are supported. The following table lists the available shortcuts in both emulated character mode and CHARVA.

shortcut	action	supported by the emulated character mode	supported by CHARVA
CTRL+Z	Undo	Yes	Yes
ATL+Backspace	Undo	Yes	No
CTRL+X	Cut	Yes	Yes
SHIFT+Delete	Cut	Yes	No
CTRL+V	Paste	Yes	No
SHIFT+INS	Paste	Yes	No
CTRL+C	Copy	Yes	No
CTRL+INS	Copy	Yes	No

More information

For more information about CHARVA, please visit: <http://www.pitman.co.za/projects/charva/>