FC_TextDisplay

Olivier LAVIALE 2004

COLLABORATORS

	TITLE : FC_TextDisplay	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Olivier LAVIALE 2004	January 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FC_TextDisplay

1.1 Feelin : FC_TextDisplay

FC_TextDisplay

This class is not a GUI class but a powerful tool to easily print formated text.

Text can be formated like HTML pages. Yet, few commands are implemented. <i>, <b>, <u> can be used to change the style of the font to, respectively, italic, bold and undelined. Text can be aligned to the left, to the right, centered or even justified with the command <align>. The command <font> can be used to change the font's face, font's colour and font's size. The command
 can be used to break lines (as well as the common \n). An customizable horizontal rule can be used to divided parts of a text with the command <hr>. Images can also be incorporated within the text with the command <img>.

In addition to standard HTML commands, some command have been added to allow further customization of the text aspect. The command <pens> allows management of 4 additionnal pens used to produce effects such as emboss, ghost, glow... The command <spacing> can be used to modify the spacing (in pixels) between two or several lines.

METHODS

FM_TextDisplay_Setup FM_TextDisplay_Cleanup

FM_TextDisplay_Draw

ATTRIBUTES

FA_TextDisplay_Contents FA_TextDisplay_PreParse

FA_TextDisplay_Font

FA_TextDisplay_Height FA_TextDisplay_Width

FA_TextDisplay_Shortcut

1.2 FC_TextDisplay / FM_TextDisplay_Draw

NAME

FM_TextDisplay_Draw -- (04.00)

SYNOPSIS

F_Do(Obj,FM_TextDisplay_Draw,FRect *Rect);

FUNCTION

Use this method to render text. If text doen't fit in the available display space, lines are truncated and three dots added e.g. 'Hel...'. If even dots don't fit nothing is displayed.

INPUTS

Rect (* FRect)

This rectangle describes the region where the text is allowed to be drawn. If text is longer then available space it's truncated and three dots "..." are drawn instead of characters that cannot fit in the available space. If (FRect -> x2 < FRect -> x1) or (FRect -> y2 < FRect -> y1) then nothing is drawn.

EXAMPLE

void drawtext(FObject TD,WORD x,WORD y,UWORD w,UWORD h) { FRect r;

r.x1 = x; r.x2 = x + w - 1; r.y1 = y; r.y2 = y + h - 1;

F_Do(TD,FM_TextDisplay_Draw,&r); }

RESULT

The method returns the width of the biggest line drawn.

SEE ALSO

FA_TextDisplay_PreParse FM_TextDisplay_Width

1.3 FC_TextDisplay / FM_TextDisplay_Setup

NAME

FM_TextDisplay_Setup -- (04.00)

SYNOPSIS

F_Do(Obj,FM_TextDisplay_Setup,FRender *Render);

FUNCTION

You need to setup the object before drawing can be made. Use this method to setup the object.

When the object is setup, text is parsed, lines and chunks are created, fonts opened, colors allocated, images created and rendered...

INPUTS

Render (FRender *)

A FRender structure needed to obtain environnement information and to perform renderings.

SEE ALSO

FM_TextDisplay_Cleanup

1.4 FC_TextDisplay / FM_TextDisplay_Cleanup

NAME

FM_TextDisplay_Cleanup -- (04.00)

SYNOPSIS

F_Do(Obj,FM_TextDisplay_Cleanup);

FUNCTION

Send this method to the object to free its ressources.

SEE ALSO

FM_TextDisplay_Setup

1.5 FC_TextDisplay / FA_TextDisplay_Contents

NAME

FA_TextDisplay_Contents -- (04.00) [ISG], STRPTR

FUNCTION

Text is formated using Tags, they delimit formating elements. Most elements are identified in a document as a start-tag, which gives the element name and attributes, followed be the content, followed be the end-tag. Start-tags are delimited by '<' and '>'; end-tags are delimited by '</' and '>'. An example is:

Text is bold Text is not bold.

Some elements only have start-tag without end-tag. For example, to create a line break, use the '
' tag.

In a start-tag, white space and attributes are allowed between the element name and the closing delimiter. An attribute specification typically consists of an attribute name, an equal sign, and a value. White space is allowed around the equal sing. In my implementation, tag names and attribute names are case sensitive, they must be in lower case.

In this example, 'img' is the element name, 'src' is the attribute name, and 'myface.png' is the attribute value:

<img src="myface.png">

b, i, u - Font style elements

These all require start and end tags e.g.

This has some bold text.

Text level elements must be properly nested - the following is an error:

This has some bold and italic text.

'b' bold text style. 'i' italic text style. 'u' underlined text style.

align - line alignement

Requires start and end tags. This allows you to change the alignment of the enclosed text. Possible values are:

left: text lines are rendered flush left. center: text lines are centered. right: text lines are rendered flush right. justify: text lines are justified to both margins.

<align="center">Hello World !</align>

br - line break

Used to force a line break. This is an empty element so the end tag is forbidden. For example:

Pease porridge hot
 Pease porridge cold
 Pease porridge in the pot
 Nice days old.

font - text font

Requires start and end tags. This allows you to change the font size and/or color for the enclosed text. Font sizes are given in terms of a scalar range or pixel size.

size

This sets the font size for the contents of the font element. You can set size to an integer for an absolute font size e.g. 'size=12px', or specify a relative percent value e.g. 'size=150%' or 'size=50%'.

color

Used to set the color to stroke the text. Colors are given as RGB in hexadecimal notation.

face

Name of the font that should be used to render text. e.g 'face="helvetica"'

hr - Horizontal Rule

The 'hr' element is a divider between sections of text; typically a full width horizontal rule or equivalent graphic. For example:

<hr> Text between lines <hr>

The ’hr’ element is not a container so the end-tag is forbidden. The attributes are:

align

This determines whether the rule is placed at the left, center or right of the space between the current left and right margin for ’align=left’, ’align=center’ or ’align=right’ respectively. By default, the rule is centered.

noshade

The attribute the rule to be rendered in a solid color rather than the traditional two colour "groove". ’noshade=true’ or ’noshade=yes’ disable the "groove" style. The solid color used can be defined with the ’shine’ attribute. By default, the rule is shaded (’noshade=false’ or ’noshade=no’).

shadow

The shadow pen used to draw the shadow part of the "groove" style. Use one of the defined color scheme pens : ’shine’, ’halfshine’, ’fill’, ’halfshadow’, ’shadow’, ’halfdark’, ’dark’, ’text’ and ’highlight’. By default the rule’s shadow pen is ’shadow’ (’shadow=shadow’).

shine

The shine pen used to draw the shine part of the "groove" style. Use one of the defined color scheme pens : ’shine’, ’halfshine’, ’fill’, ’halfshadow’, ’shadow’, ’halfdark’, ’dark’, ’text’ and ’highlight’. By default the rule’s shine pen is ’shine’ (’shine=shine’).

size

This can be used to set the height of the rule in pixels. You can use decimal values (e.g. ’size=3’) as pixel values (e.g. ’size=3px’). By default the rule is 2 pixel high (’size=2px’).

width

This can be used to set the width of the rule in pixel (e.g. ’width=100’ or ’width=100px’) or as the percentage between the current left and right margins (e.g. ’width=50%’). The default is 100% (’width=100%’).

img - inline images

Used to insert images. ’img’ is an empty element and so the end tag is forbidden. ’img’ elements support the following attributes:

src

This attribute is required for every ’img’ element. It specifies the address of the image resource (e.g. "images/icon.png").

height

Specifies the intended height of the image in pixels. If the value is not defined, the image’s height is obtained from the image object.

width

Specifies the intended width of the image in pixels. If the value is not defined, the image’s width is obtained from the image object.

pens

Requires start and end tags. This allows you to manage text pen and 4 addition pens to create stylish text effect such as emboss, glow... Don’t mix up pen and colour, especially for text. A pen is one of the available colours from the Feelin’s color scheme, which are ’shine’, ’halfshine’, ’fill’, ’halfshadow’, ’shadow’, ’halfdark’, ’dark’, ’text’ and ’highlight’. In other words, a pen is an abstract reference and a colour is an RGB value. The following attributes are defined:

up

Use this attribute to define the pen to use to draw text at ’text_x - 1’ ’text_y - 1’ over the standard text.

down

Use this attribute to define the pen to use to draw text at ’text_x + 1’ ’text_y + 1’ over the standard text.

light

Use this attribute to define the pen to use to draw text at ’text_x - 1’ ’text_y - 1’ behind the standard text.

shadow

Use this attribute to define the pen to use to draw text at 'text_x + 1' 'text_y + 1' behind the standard text.

text

Use this attribute to define the pen to use to draw text. Note: If the command is encapsulated in a ... section and a colour has been defined for this section (.e.g "<font color="#FF0000">"), defining the pen will have no effect, because the color is prioritary.

style

Use this attribute to use one of the available style that will define for you the approriate pens. The available styles are :

emboss: "text=text up=shine" ghost: "text=halfshadow shadow=halfshine" glow: "text=text light=shine shadow=halfshadow" light: "text=text light=shine" shadow: "text=text shadow=halfshadow"

stop

Disable formating engine for the enclosed section. Altough this element requires start and end tags it is not nested, you should pay attention to this.

NOTE

Note that the text suplied with the FA_TextDisplay_Contents attribute is not cloned. The string must remains valid until you set the attribute to NULL or you dispose the FC_TextDisplay object.

SEE ALSO

FA_TextDisplay_PreParse FC_Text / FA_Text

1.6 FC_TextDisplay / FA_TextDisplay_PreParse

NAME

FA_Text_PreParse -- (01.00) [ISG], STRPTR

FUNCTION

String containing format definitions to be parsed before the text from FA_TextDisplay_Contents is printed.

Using this tag, you can easily define different formats, colors and styles without modifying the original string. This string must remain valid until you set it to NULL or dispose the object.

EXAMPLE

... FA_TextDisplay_PreParse, "<align=center><i>", -> centered and italics FA_TextDisplay_Contents, "foobar", ...

1.7 FC_TextDisplay / FA_TextDisplay_Font

NAME

FA_TextDisplay_Font -- (01.00) [ISG], struct TextFont *

FUNCTION

Defines the base font to draw with.

This attribute must be defined, otherwise text won't be setuped and drawing will be impossible.

SEE ALSO

FM_TextDisplay_Draw FM_TextDisplay_Setup

1.8 FC_TextDisplay / FA_TextDisplay_Height

NAME

FA_TextDisplay_Height -- (01.00) [ISG], UWORD

FUNCTION

This attribute is use to set (limit) text's height, and to get real text's height.

NOTE

Text is adjusted, according to limits given by FA_TextDisplay_Width and FA_TextDisplay_Height, when the object is setup or one of the attributes FA_TextDisplay_Contents , FA_TextDisplay_PreParse, FA_TextDisplay_Shortcut , FA_TextDisplay_Font , FA_TextDisplay_Width or FA_TextDisplay_Height is modified. Text is also adjusted if the dimension of the rectangle given width FM_TextDisplay_Draw differs from the one given by FA_TextDisplay_Width and FA_TextDisplay_Height.

1.9 FC_TextDisplay / FA_TextDisplay_Shortcut

NAME

FA_TextDisplay_Shortcut -- (01.00) [ISG], BOOL | UBYTE

FUNCTION

Set this attribute to TRUE (default) if you want the underscore character (shortcut) to be handled. Getting this attribute will return the underscored character (shortcut). Only the first underscore found will stand as shortcut.

Setting this attribute to FALSE will disable underscore handling, all underscores will be simple characters.

1.10 FC_TextDisplay / FA_TextDisplay_Width

NAME

FA_TextDisplay_Width -- (01.00) [ISG], UWORD

FUNCTION

This attribute is use to set (limit) text's width, and to get real text's width.

NOTE

Text is adjusted, according to limits given by FA_TextDisplay_Width and FA_TextDisplay_Height , when the object is setup or one of the attributes FA_TextDisplay_Contents , FA_TextDisplay_PreParse, FA_TextDisplay_Shortcut , FA_TextDisplay_Font , FA_TextDisplay_Width or FA_TextDisplay_Height is modified. Text is also adjusted if the dimension of the rectangle given width FM_TextDisplay_Draw differs from the one given by FA_TextDisplay_Width and FA_TextDisplay_Height .