text

COLLABORATORS			
	TITLE : text		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 31, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

text

1.1 text_plugin: Introduction

```
                text_plugin

    by Ali Graham <agraham@hal9000.net.au>


text_plugin is a simple equivalent to the
TEXT gadget in EasyGUI, that can also use a different
font than the window is using. It also features the
justification of the text within the gadget to the
left, right or middle.

------------------------------------------------------------


              Constructor


              Methods


              Tags


              Exceptions


              History
```

1.2 text_plugin: Constructors / Destructor

```
                Constructor

    text(
              tags
              :PTR TO tagitem)
```

For creating a new plugin object use for example:

```
DEF text:PTR TO text_plugin
NEW text.text([..., TAG_DONE])
```

1.3 text_plugin: New Methods

```
                set(
              tag
              , value)
```

By calling this method it's possible to change attributes
at runtime. You can use all tags with the S flag set.
This method can also be used before the GUI is created and when
the window is closed.
Value is a LONG and contains the argument for the used tag.

Example:

```
DEF text:PTR TO text_plugin
...
NEW text.text([..., TAG_DONE])
...
text.set(PLA_Text_Disabled, TRUE)
...
```

```
value,check:=get(
              tag
              )
```

This method is the counterpart to set. All tags with G flag
can be used. Argument is the tag you want to get. Return values
are the requested value and as second a boolean value. So if
check is FALSE the used tag can't be get.

```
...
value,check:=text.get(PLA_Text_Disabled)
...
```

After this:

```
 value=TRUE
 check=TRUE
```

But if you try:

```
    ...
    value,check:=text.get(PLA_Text_Font)
    ...

    Then you get this:
     value:=-1
     check:=FALSE


draw()              /* Private */
```

1.4 text_plugin: Tags

```
                The letters [ISG] show you when the tags can be used.
I =
               Initalisation
                S =
               Set Method
                G =
               Get Method
                   PLA_Text_Text           [I.G]

      The contents of the text field.


  PLA_Text_Highlight      [ISG]

      Boolean; whether or not the text should
      be rendered highlighted (in white) instead
      of normally (in black).


  PLA_Text_ThreeD         [ISG]

      Boolean; whether or not the text should
      be rendered with a 3D look. Defaults to
      FALSE (and, when TRUE, will look different
      depending on whether or not PLA_Text_Highlight
      is set).

  PLA_Text_DrawBar        [ISG]

      Boolean; whether or not to draw a bar to the
      left and right of the text, to achieve a look
      similar to the old title_plugin.
      Defaults to FALSE.


  PLA_Text_Font           [I.G]

      The font that the text field should use. This
      is a pointer to a textattr structure which
```

represents an available font. Default is NIL;
this means that the PLUGIN will use the
window's font.

PLA_Text_Justification [ISG]

Set the justification of the text within the
available space.

Can be set to one of three values:

PLV_Text_JustifyCenter
PLV_Text_JustifyLeft
PLV_Text_JustifyRight

PLA_Text_Disabled [ISG]

Disable or enable a gadget. Setting this tag causes the
gadget to become disabled; it gets a ghost pattern.

1.5 text_plugin: Exceptions

Constructor

"util" will be raised if the utility.library has not
been opened.

1.6 text_plugin: History

v1.0 (2.10.97)

o Initial release.

v1.1 (28.10.97)

o Added text justification (left, right & center).

o Some sources using this module may need to be changed;
the calling parameters of the initial method have been
modified.

o Removed some unnecessary code from the render() method.

v1.2 (28.11.97)

o Rewritten to follow Ralph Wermke's PLUGIN Style Guide.

- Added the ability to disable the text field.

- Changed 3D mode, and added ability to draw bars (duplicates the functionality of the now defunct title_plugin).