Table of Contents

! <
- ; <
<
%# !" <
= . / ) > %"
! <
- ; <
<
%# !" <
= . / ) ! > %"
! <
- ; <
<
%# !" <
= > %"
! <
- ; <
<
%# !" <
= > %"
! <
- ; < 1&
<
%# !" <

1&

) " ? 3
) " ? ; 3
) " ? ;

pixels.

Sti

where:

- $s$ is the scale value
- $o$ is the pixel o

```
{{sx,ox},{sy,oy}}
```

wh

```

```

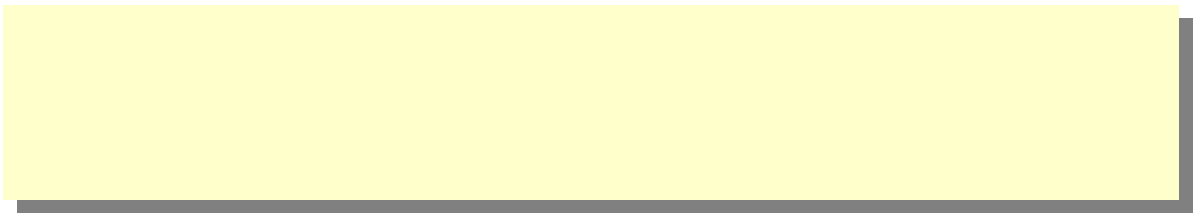
```

```

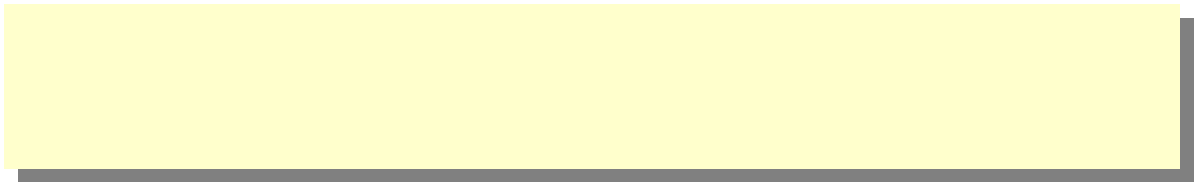
```

```

*Prop*



section.



```
// Create a new widget
wMgr.createWindow("FunkyLook/Button", "myFunkyButton");
```

Here we create an instance of the new widget, and name it "myFunkyButton". The widget can now be attached to other windows and generally used as you would any 'normal' widget.

## *Conclusion*

This concludes the overview of the new parts of the CEGUI system.

You have seen how the new ªUnifiedº co-ordinate system works, and how to make use of the new window alignment options.

You have also learned the basics of how to set up your scheme files to initialise the Falagard base module, and how to map XML defined skins to the base Falagard widgets to cre

To explain this further, the example from the reference section will be used. Basically, in that example we want to take the height of the widget font, add four pixels and multiply the result by two. This mi0Òlead us to write the following, wronaspe

```
...
<FontDim type=ºLineSpacinaº
```

&

Falagard XML Referen

*<Area> Element*

## *<AreaProperty> Element*

*Purpose:*

The <AreaPropertyèOÒ element is intended to allow the system to access a propertp on the target window to obtain the final target area of a component being defined.

*Attribetes:*

- name ± specifies the name of the propertp to access. The named propertp must acc fisfis VPhuRfiquired atGibute.

^ *sage:*

g te.

- The <AreaPropertyelementÒ#0te

te.

<Child> ElemQUQ~tBO<ChD

## *<Dim> Element*

*Purpose:*

The <Dim> element is intended as a container element for a single di

## *<DimOperator> Element*

*Purpose:*

The <DimOperator> element allows you to combine two of the specialised dimension specifier elements via a simple mathematical operator. Since the dimension used as the second

## *<Falagard> Element*

*Purpa%W'se*

The

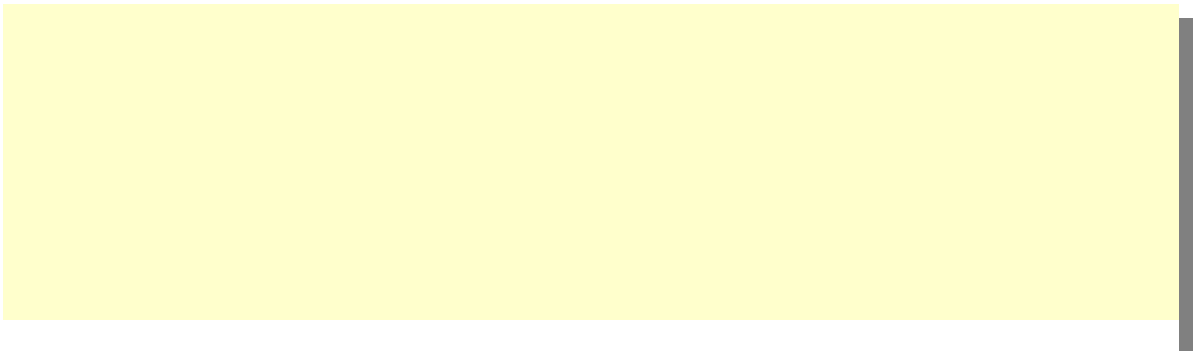
## *<FontDim> Element*

*Purpose:*

## *<ImageProperty> Element*

*Purpose:*

The <ImageProperty>



## *<StateImagery> Element*

*Purpose:*

The <StateImagery> element de

## *<VertAlignment> Element*

*Purpose:*

<VertFormat> Eld@D¥DfcÜG'DÝVqmÐ<VrtFf0s¥Do

## *<VertFormatProperty> Element*

*Purpose:*

The <VertFormatProperty> element is intended to allow the system to access a property on the target

# Falagard Base Widgets Reference

## *Falagard/Button*

General

## *Falagard/Combobox*

Widget providing a text box and drop down list invoked via a button.

Assigned WidgetLook should provide the following:

- StateImagery definitions:
  - Enabled ± General imagery for when the widget is enabled.
  - Disabled ± General imagery for when the widget is disabled.

- Child widget definitions:
  - Editbox based widget with name suffix ª__auto_editbox__º
  - ComboDropList based widget with name suffix ª__auto_droplist__º
  - Pushé

Falagard/Editbox

General purpose singEe-Eine`Xbox widget.

Assigned WidgetLook sg@

- DisabledWithTitleNoFrame ± Imagery used when the widget has its title bar enabled, has its frame disabled, and is disabled.
- ActiveNoTitleWithFrame ± Imagery used when the widget has its title bar disabled,
-

- NamedArea definitions:
  - ItemRenderingArea ± Target area where list items will appear when no scrollbars are visible (also acts as default area). Required.
  - ItemRenderingAreaHScroll ± Target area where list items will appear when the horizontal scrollbar is visible. A onal.
  - ItemRenderingAreaVScroll ± Target area where list items will appear when the al scrollbar is visible. @ ional.
  - ItemRenderingAreaHVScr

- Normal ± Imagery to use when the widget is enabled and the mouse is not within any part of the segment widget.
- Hover ± Imagery to use when the widget is enabled and the mouse is within the main area of the wig@ (not the drag-sizing ‹splitter› area).
- SplitterHover ± Imagery s Ø e when x e widge

s

Assigned WidgetLook should provide the following:

- StateImagery definitions:
  - EnabledNormal ± Imagery used when the item is enabled and the mouse is not within its area.
  - EnabledHover ± Imagery used when the item is enabled and the mouse is within

- Property initialiser definitions:
  - SelectionBrushImage ± defines name of image that will be painted for the text selection (this is applied on a per-line basis).

- ReversedProgress ± boolean property.  Determines whether the progress

## *Falagard/StaticImage*

Static widget that displays a configurable image.

Assigned WindowRenderer should provide the following:

- StateImagery definitions:
  - Enabled ± General imagery for when the widget is enabled.
  - Disabled ± General imagery for when the widget is disabled.

- NamedArea definitions:
  - TextRenderArea ± Target area where text will appear when no scrollbars are visible (also acts as default area). Required.
  - TextRenderAreaHScroll ± Target area where text will appear when the horizontal scrollbar is visible. Optional.
  - TextRenderAreaVScroll ± Target area

Assigned WidgetLook shou