

Table of Contents

```

-      ;      <
      <
%#      !"      <
=      "      >      %"
      !      <
-      ;      <
      <
%#      !"      <
=      "      !      >      %"
      !      <
-      ;      <
      <
      Đ

```

j

j

j

j

```

'0
'0
'0
'1
'1
'1
'1
'1
'1
'8
'8
'8
'8
'8

```

j

	!	<			0
	-	;	<		0
			<		0
	%#	!"	<		0
=.	/)		>	%"	0
	!	<			0
	-	;	<		0
			<		0
	%#	!"	<		0
=.	/)		!	>	%"
	!	<			1\$
	-	;	<		1\$
			<		1\$
	%#	!"	<		1\$
=			>	%"	1
	!	<			1
	-	;	<		1
			<		1
	%#	!"	<		1
=			>	%"	1&
	!	<			1&
	-	;	<		1&
			<		1&
	%#	!"	<		1&

pixels®

Sti



where:

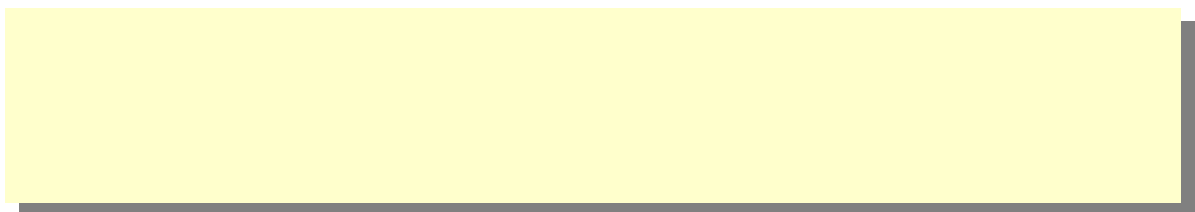
- s^o is the scale value
- o^o is the pixel o



$\{\{sx, ox\}, \{sy, oy\}\}$

wh

Prop



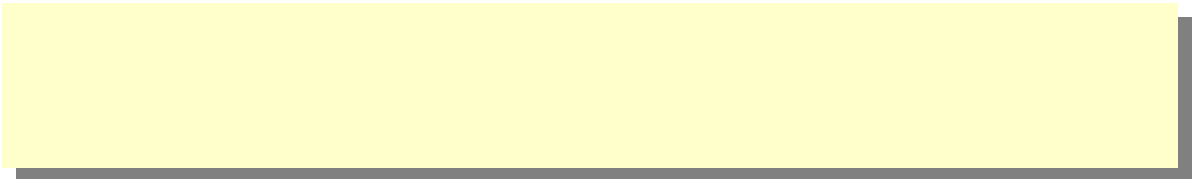


[Redacted]

[Redacted]

[Redacted]

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.

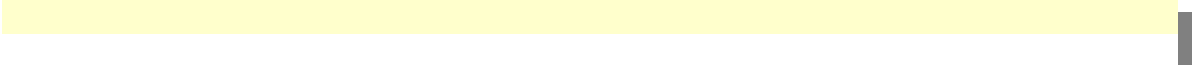
Conclusion

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

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

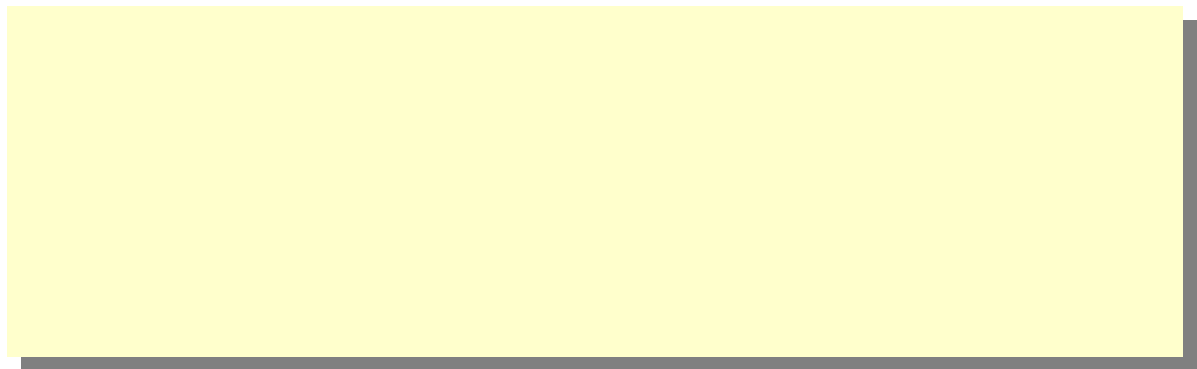
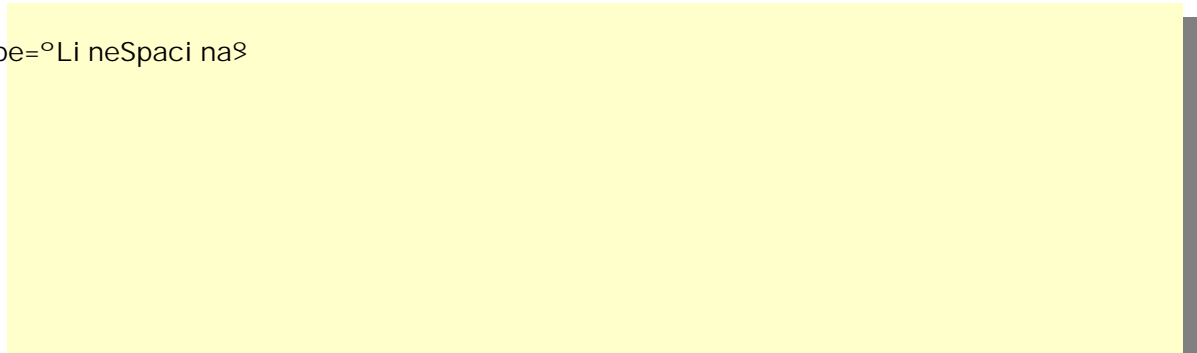
You have also learned the basics of how to set up your scheme files to initialise the Falagard module, and how to map XML defined skins to the base Falagard widgets to create

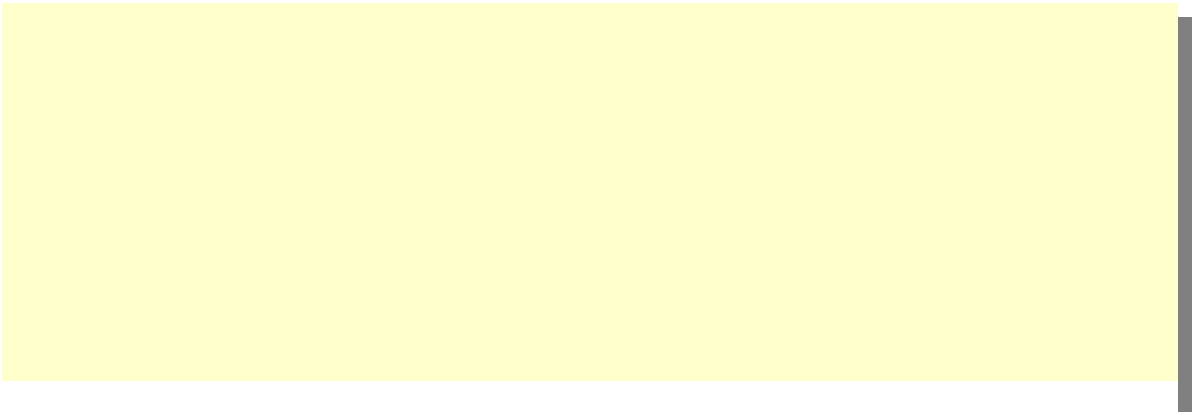




To explain this further, the example from the reference section will be used. Basically, in that example we want to take the height of the widget font, add four pixels and multiply the result by two. This will lead us to write the following, wrong aspe

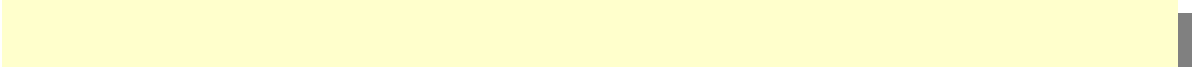
```
...  
<FontDim type="LineSpacing"
```



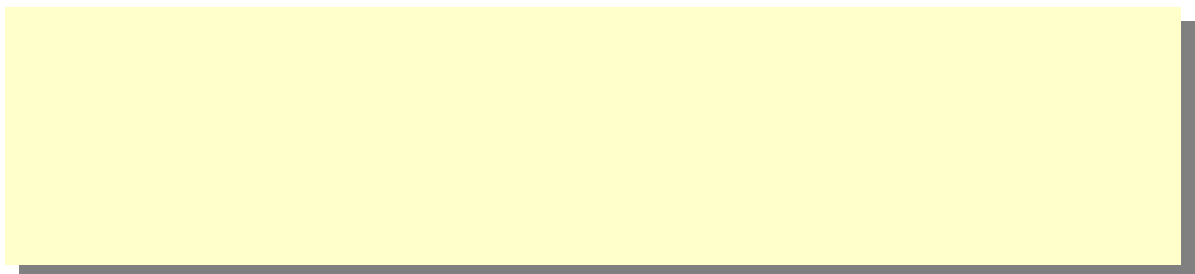


Falagard XML Referen





<Area> Element



<AreaProperty> Element

Purpose:

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

Attributes:

- name ± specifies the name of the property to access. The named property must access a value required at runtime.

Usage: The <AreaProperty> element is used to define the target area of a component.

- The <AreaProperty> element is used to define the target area of a component.



<Child> ElemQUQ~~~0~~0<ChD



<Dim> Element

Purpose:

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



<DimOperator> Element

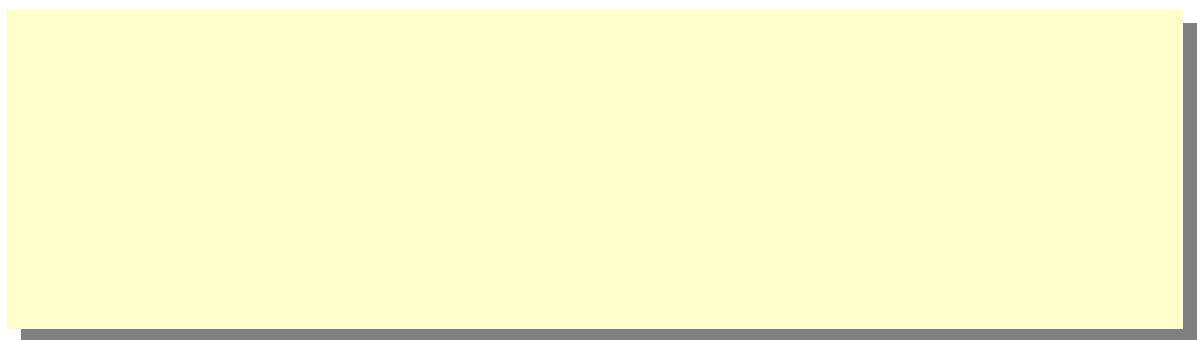
Purpose:

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

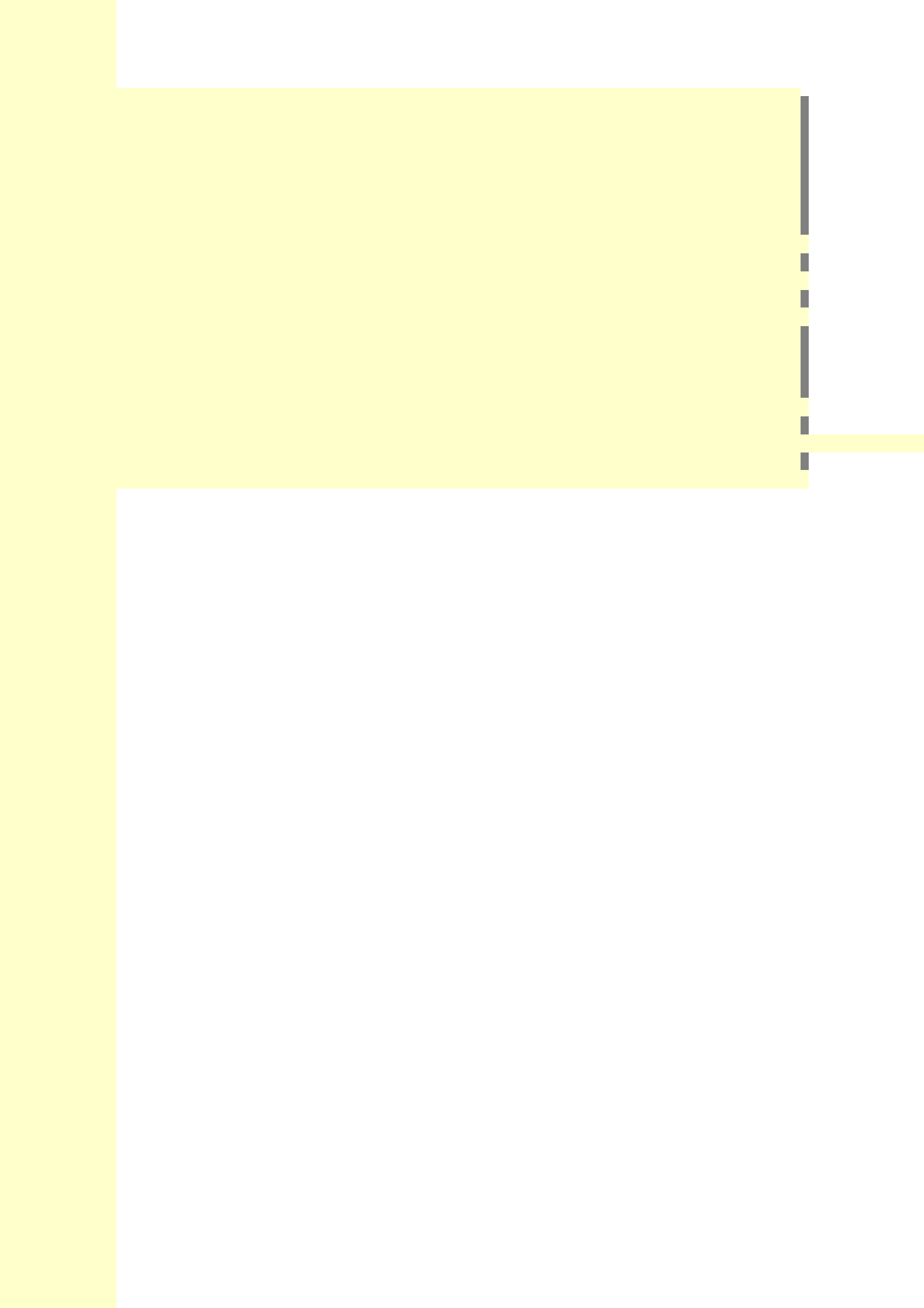
The

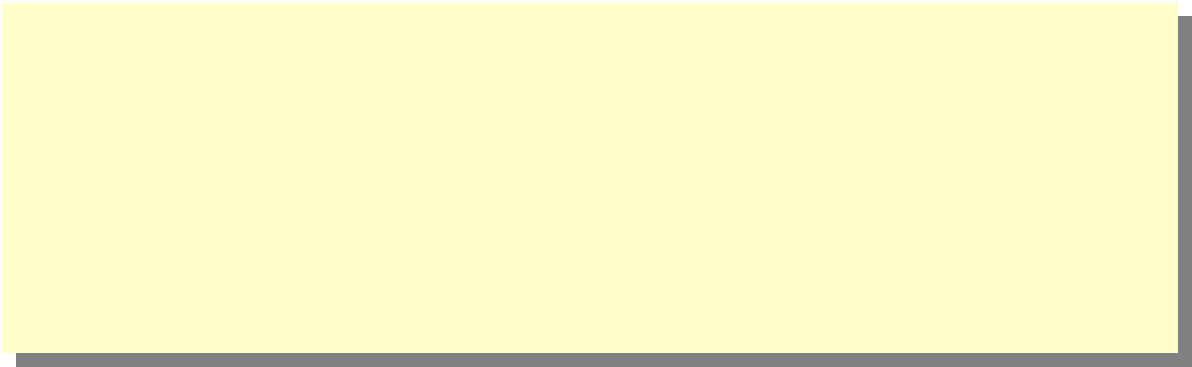


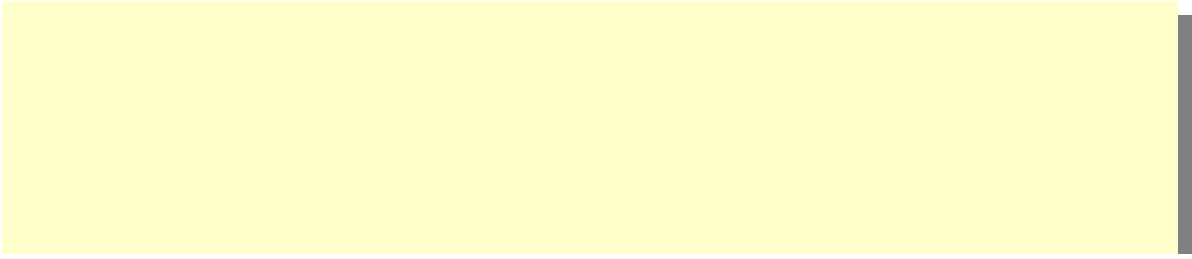
<FontDim> Element

Purpose:







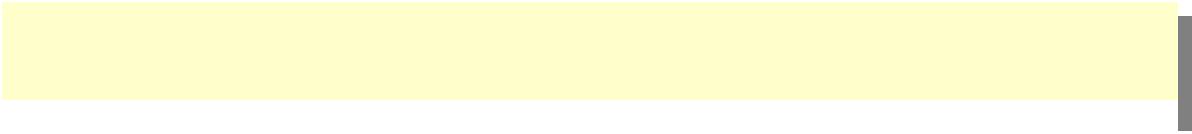


<ImageProperty> Element

Purpose:

The <ImageProperty>





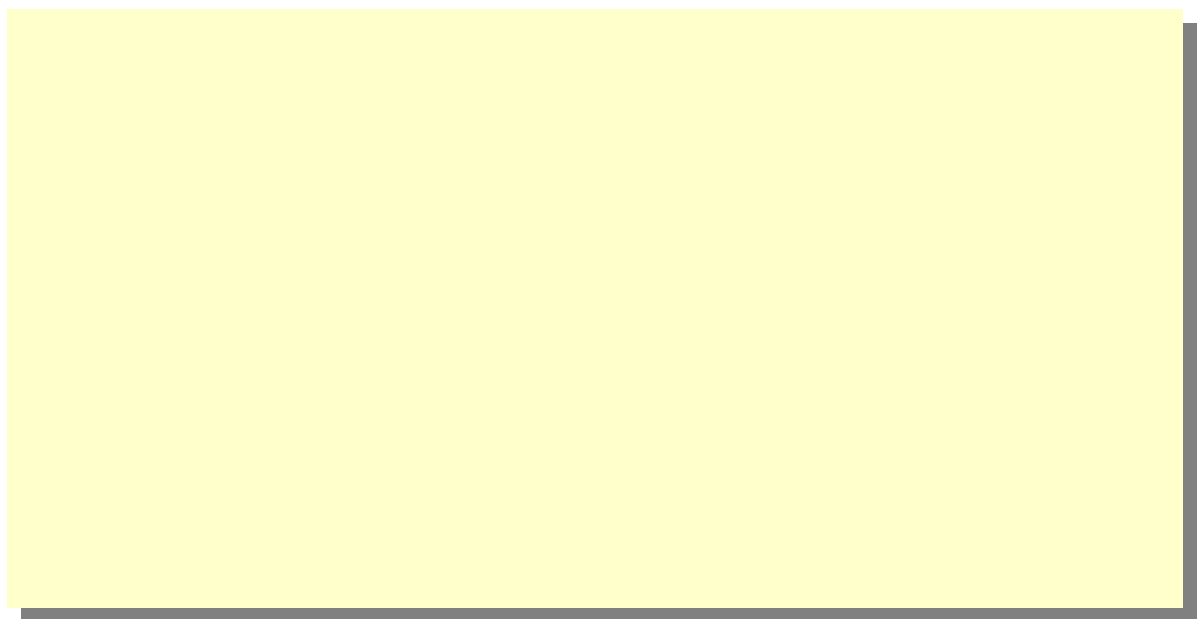


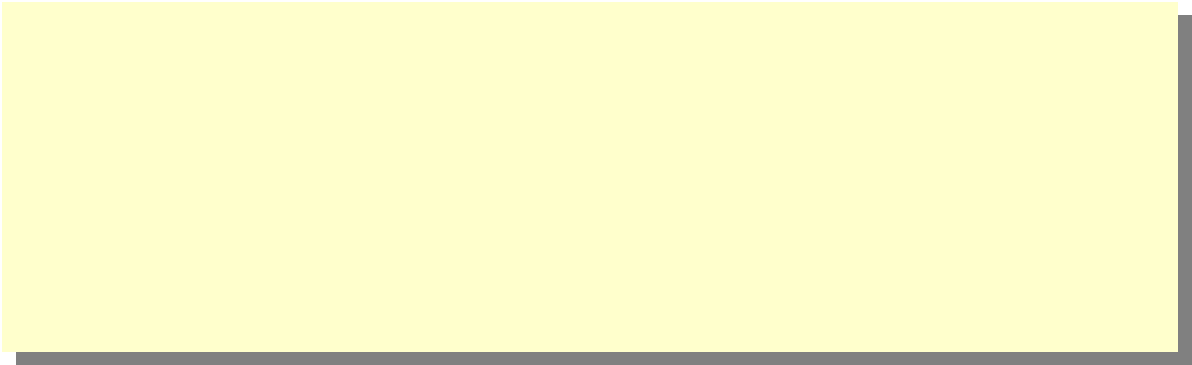


<StateImagery> Element

Purpose:

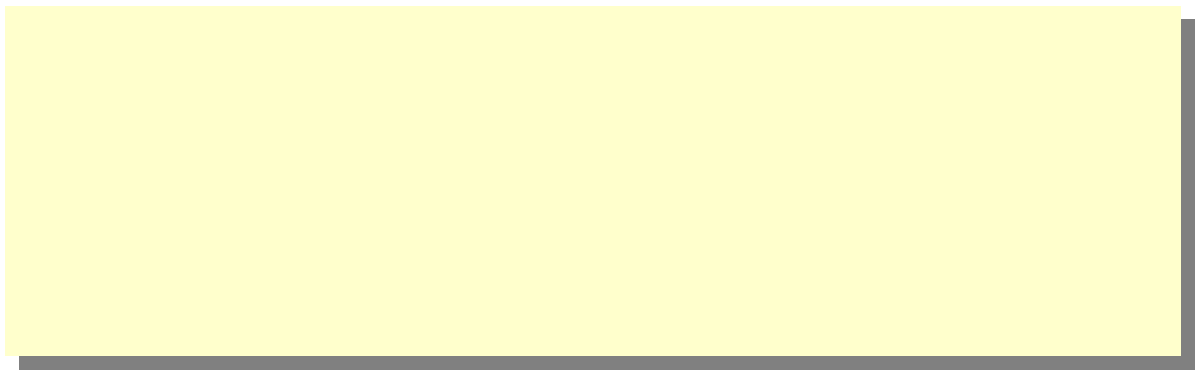
The <StateImagery> element de



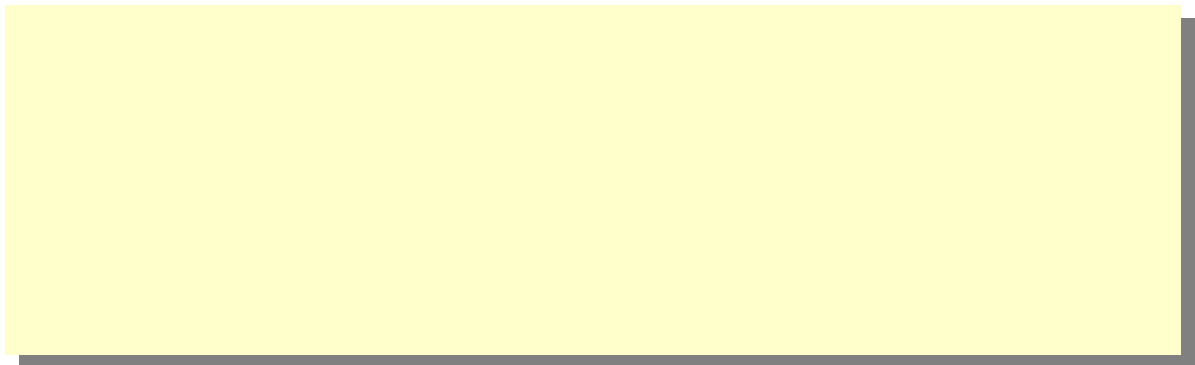
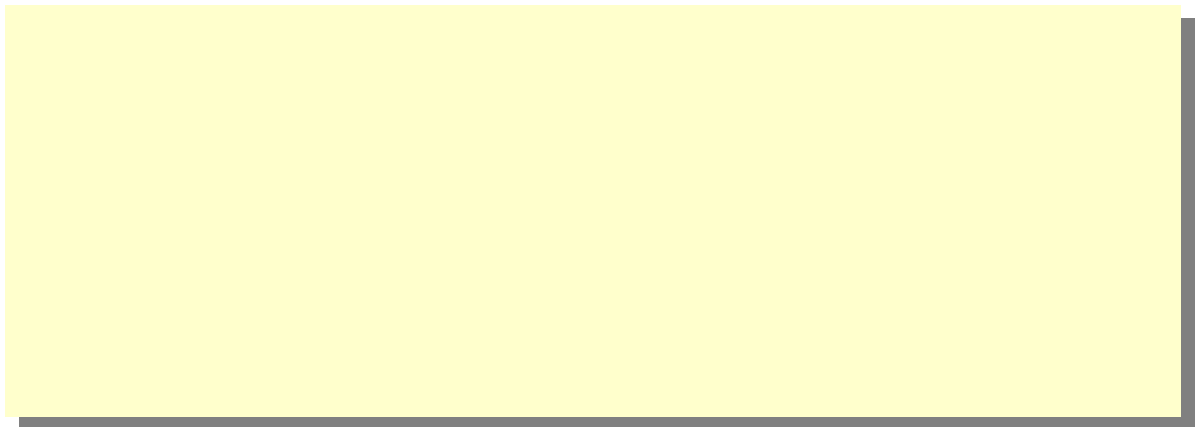


<VertAlignment> Element

Purpose:



<VertFormat> Eld@D'DfcDG'DMqmD<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
 - Push

Falagard/Editbox

General purpose singEe-Eine ~~xx~~ box 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,
- Icon

gta pÖB x•

!V@

Dc c

lg#Rv2e-fP

- 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. Optional.
 - ItemRenderingAreaVScroll ± Target area where list items will appear when the vertical scrollbar is visible. Optional.
 - 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 widget (not the drag-sizing splitter area).
- SplitterHover ± Imagery to use when the widget is enabled and the mouse is within the splitter area.

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 \pm boolean property. Determines whether the progress

