RuledScrollView

INHERITS FROM	ScrollView : View : Responder : Object

CLASS DESCRIPTION

The RuledScrollView class is a general purpose class designed to allow 'rulers' to be attached along the edge of a document view within a ScrollView.   The movement of these "ruler views" is constrained, such that a horizontal ruler along the bottom or top of the document view scrolls horizontally along with the document view, but does not move vertically in the ScrollView.   A vertical ruler along the left or right edge is similarly constrained to only scroll vertically.   Controls such as buttons or a PopupList may be imbedded in the scrollers of the RuledScrollView by simply connecting outlets in Interface Builder.

Currently, the views that are used as ruler views with a RuledScrollView must be a member of the class Ruler.   This is because the size of a ruler is set to correspond to the size of the docView in

RuledScrollView's setSize method, and this method invokes the Ruler setSize method. In the next version, I plan to change this so that RuledScrollView does all the resizing itself, giving more freedom to what can be used as a ruler view. (In the meantime, if you really don't want to make your ruler view a subclass of Ruler, you could probably just copy Ruler's setSize method to your class.)

EXAMPLE

The example program "BrainRulers" should have been included together with this class. This example gives a fairly good idea of what the class does and how it is used.

LIMITATIONS/BUGS

· Ruler views must be a subclass of Ruler, or the setSize method must be copied to the class.
· StubViews do not show up on a printout, or in the eps image generated by writeToStream: (which can be easily used to provide drag and drop, as illustrated in the example). The addition of stub views was made after the writeToStream: method was written, and it hasn't been updated yet...
· Code to tile StubViews when the primaryRulers are the left/right rulers is still missing. This would be a simple addition to the existing tile method, but I haven't had a desire for left/right primary rulers in any of my applications, so I haven't gotten around to this yet.
· The documentation for this class is incomplete. In case of incompleteness, please refer to the code:-)

· I plan to make an IB palette with this class and the Ruler class, but I think I'll wait for 3.0.

This is Version 0.9 of RuledScrollView, released August 1992.   Please contact the author to see if there is a more up-to-date version available.   I am "releasing" this not-quite-complete version because

Author:
Kevin Brain (ksbrain@zeus.uwaterloo.ca)
University of Waterloo / Department of Systems Design / Waterloo, Ontario/N2L 3G1

Based on the TileScrollView class and ScrollDoodScroll example by Jayson Adams, NeXT Developer Support Team

INSTANCE VARIABLES

Inherited from Object	Class	isa;

Inherited from Responder	id	nextResponder;
Inherited from View	NXRect	frame;
	NXRect	bounds;
	id superview;	
	id subviews;	
	id window;	
	struct __vFlags	vFlags;
Declared in ScrollView	id	vScroller;
	id hScroller;	
	id contentView;	
	float	pageContext;
	float	lineAmount;
Declared in RuledScrollView	View	*mainView
	View	*printView
	id leftRuler	
	id rightRuler	
	id topRuler	
	id bottomRuler	
	id bottomLeftStub	
	id topLeftStub	

```
id    bottomRightStub
id    topRightStub
ClipView                   *leftRulerClipView
ClipView                   *rightRulerClipView
ClipView                   *topRulerClipView
ClipView                   *bottomRulerClipView
NXRect                     oldMainClipRect
NXRect                     oldLeftRect
NXRect                     oldRightRect
NXRect                     oldTopRect
NXRect                     oldBottomRect
NXRect                     oldRect
id    hScrollerLeftEmbeddedView
id    hScrollerRightEmbeddedView
id    vScrollerTopEmbeddedView
id    vScrollerBottomEmbeddedView
BOOL                       rulersOn
int   rulerVisible[4]
id    printWindow
ClipView                   *mainPrintClipView
NXPoint                    mainVisiblePoint
int   primaryRulers
```

mainView	the docView, used during printing.
printView	view used to construct image to be printed.
leftRuler	
rightRuler	
topRuler	
bottomRuler	
bottomLeftStub	
topLeftStub	
bottomRightStub	
topRightStub	
leftRulerClipView	ClipView holding left ruler.
rightRulerClipView	ClipView holding right ruler.
topRulerClipView	ClipView holding top ruler.
bottomRulerClipView	ClipView holding bottom ruler.
oldMainClipRect	original rect of Main ClipView before adjusting it for printing
oldLeftRect	original rect of left ClipView before adjusting it for printing
oldRightRect	original rect of right ClipView before adjusting it for printing
oldTopRect	original rect of top ClipView before adjusting it for printing
oldBottomRect	original rect of bottom ClipView before adjusting it for printing
oldRect	size of docView the last time tile was invoked
hScrollerLeftEmbeddedView	view embedded in horizontal scroller
hScrollerRightEmbeddedView	view embedded in horizontal scroller
vScrollerTopEmbeddedView	view embedded in vertical scroller

vScrollerBottomEmbeddedView	view embedded in vertical scroller
rulersOn	whether rulers are to be displayed
rulerVisible[4]	whether each ruler is on or off
printWindow	
mainPrintClipView	holds the mainView while constructing the printView
mainVisiblePoint	point to scroll mainView to after print reconstruction
primaryRulers	which rulers extend to edge

METHOD TYPES

Initializing and freeing an instance	- initFrame:(NXRect *)frameRect;
	- free;
	// - awake;
IB Custom Palette Support	//- (const char*)inspectorName;
	//- read:(NXTypedStream *) s;
	//- write:(NXTypedStream *) s;
Adding views connected in IB	- setLeftRuler:
	- setRightRuler:
	- setTopRuler:
	- setBottomRuler:
	- setBottomLeftStub:

	- setTopLeftStub:
	- setBottomRightStub:
	- setTopRightStub:
Adding views programmatically	- addRulerView: toEdge:
	- addStubView: toCorner:
Maintaining proper scrolling behavior	- tile
	- reflectScroll:
	- scrollClip:to:
	- setSizeIfNeeded
	- setSize
Setting/getting primary rulers	- setPrimaryRulers:
	- primaryRulers
Hiding/displaying rulers	- showRuler:
	- hideRuler:
	- isRulerVisible:
Returning rulers and stubs	- topRuler
	- bottomRuler
	- leftRuler

	- rightRuler
	- bottomLeftStub
	- topLeftStub
	- bottomRightStub
	- topRightStub
Getting minimum size	- getMinSize:
Printing	- printVisible:
Writing view to stream	- writeToStream:

INSTANCE METHODS

addRulerView:toEdge:

- **addRulerView:**(Ruler *)*theView* **toEdge:**(int)*edge*

Adds *theView* as the ruler along the edge given by *edge*. The value of *edge* is one of LEFTEDGE, BOTTOMEDGE, RIGHTEDGE or TOPEDGE, which are defined in RuledScrollView.h. (Note: These manifests also correspond to the slice parameter of the NXDivideRect function.) If *edge* already has a ruler, that ruler and its ClipView are freed.

addStubView:toCorner:

- **addStubView:**(View *)*theView* **toCorner:**(int)*corner*

Adds *theView* as the stub view in the corner given by *corner*. The value of *corner* is one of BOTTOMLEFTCORNER, TOPLEFTCORNER, BOTTOMRIGHTCORNER or TOPRIGHTCORNER, which are defined in RuledScrollView.h. If *corner* already has a stub view, it is freed.

free

- **free**

Frees all disposable storage used by the RuledScrollView. Returns [super free].

initFrame:

- **initFrame:**(NXRect *)*frameRect*

Initializes and returns the receiver, a new RuledScrollView instance. The value of *rulersOn* is set to YES, *primaryRulers* is set to TOPBOTTOM, and the rulers themselves are set to NULL.

setSize

- **setSize**

Sets the sizes of the rulers and stub views according to the current setup. Invokes setSize for each ruler then gets the views to be properly retiled (positioned within the ScrollView) by sending resizeSubviews: to itself. This method must be invoked whenever the size of the mainView is changed, or when rulers or stubviews are added or removed. Returns self.

See also: **± setSizeIfNeeded**

setSizeIfNeeded

- setSizeIfNeeded

This method compares the current size of the docView with the size it was the after the time **tile** was invoked. If the sizes differ, setSize is invoked. If you do something in your program that may or may not have changed the size of the docView, you may call this, and setSize will be performed if necessary. Returns self.

See also: **± setSize**