Mike Ferris - October 9th, 1993
Georg Tuparev - April 17th, 1995

MiscMatrix

Inherits From: Matrix : Control : View : Responder : Object

Declared In: misckit/MiscMatrix.h

Class Description

MiscMatrix is a subclass of Matrix which allows its rows and columns to be sized independantly. Each row and/or column can have a different height or width.

The methods -setWidth:ofCol: and -setHeight:ofRow: are used to set up the matrix. When the object is initialized, and as new rows or columns are added, the cellSize (inherited from Matrix) is used for the initial size. After the cells have been added, the above methods are used to resize them to the correct size.

Instance Variables

id **columnSizes**;

id **rowSizes**;

columnSizes	A Storage object of MiscColumnSize structures.
rowSizes	A Storage object of MiscRowSize structures.

Method Types

Initializing the class	+initialize
Creating and freeing instances	- initFrame:mode:prototype:numRows:numCols: - initFrame:mode:cellClass:numRows:numCols: - copyFromZone: - free
Cell sizes	- setWidth:ofCol: - setHeight:ofRow: - widthOfCol: - heightOfRow: - totalWidthPerRow:
Laying out the matrix	- numCols - numRows
Overridden methods	- sizeToCells - sizeToFit

- setAutosizeCells:
- renewRows:cols:
- insertColAt:
- insertRowAt:
- removeColAt:
- removeRowAt:
- drawSelf::
- getCellFrame:at::
- getRow:andCol:forPoint:
- setIntercell:

Archiving - read:
- write:

Class Methods

initialize
+ **initialize**

Initializes the class instance doing things like setting the version number.

Instance Methods

copyFromZone:

- **copyFromZone:**(NXZone *)*zone*

Makes a copy of the matrix.   This method needs to make a copy of the storage objects.

See also: ±€**initFrame:mode:prototype:numRows:numCols:**, ±€**initFrame:mode:cellClass:numRows:numCols:**, ±€**free**

drawSelf::
- **drawSelf:**(const NXRect *)*rects* **:**(int)*rectCount*

Draws the matrix.

free
- **free**

Frees the memory we may have allocated for various things.

See also: ±€**initFrame:mode:prototype:numRows:numCols:**, ±€**initFrame:mode:cellClass:numRows:numCols:**, ±€**copyFromZone:**

getCellFrame:at::
- **getCellFrame:**(NXRect *)*rect* **at:**(int)*rowNum* **:**(int)*colNum*

Calculates and returns the rectangle containing the given cell.

getRow:andCol:forPoint:
- **getRow:**(int *)*row* **andCol:**(int *)*col* **forPoint:**(const NXPoint *)*point*

Calculates and returns the row and column of the cell under the given point.   Returns -1, -1 if the point is not over any cell.

heightOfRow
- (float)**hieghtOfRow:**(int)*aRow*

Returns the height of *aRow.*

See also: ±€**widthOfCol:, ±€totalWidthPerRow**

initFrame:mode:cellClass:numRows:numCols:
- **initFrame:**(const NXRect *)*frm* **mode:**(int)*aMode* **cellClass:***factory* **numRows:**(int)*rowsHigh* **numCols:** (int)*colsWide*

Initialize our size storage.

See also: ±€**initFrame:mode:prototype:numRows:numCols:**, ±€**copyFromZone:**, ± **free**

initFrame:mode:prototype:numRows:numCols:
- **initFrame:**(const NXRect *)*frm* **mode:**(int)*aMode* **prototype:***protoCell* **numRows:**(int)*rowsHigh* **numCols:** (int)*colsWide*

Initialize our size storage.

See also: ±€**initFrame:mode:cellClass:numRows:numCols:**, ±€**copyFromZone:**, ± **free**

insertColAt:
- **insertColAt:**(int)*position*

Adds a new column at the given position, pushing other columns to the right. The new column has the default width.

insertRowAt:
- **insertRowAt:**(int)*position*

Adds a new row at the given position, pushing other rows down. The new row has the default height.

numCols
- (int)**numCols**

Returns the number of coloumns in the MiscMatrix

See also: ±€**getNumRows:numCols:** (Matrix)

numRows
- (int)**numRows**

Returns the number of rows in the MiscMatrix

See also: ±€**getNumRows:numCols:** (Matrix)

read

- **read:**(NXTypedStream *)*strm*

Reads the object from the typed stream. Overriden to read in the size storage.

See also: ±€**write:**

removeColAt:

- **removeColAt:**(int)*position*

Removes the column at the given position. Columns beyond it are moved left to fill the gap.

removeRowAt:

- **removeRowAt:**(int)*position*

Removes the row at the given position. Rows beyond it are moved up to fill the gap.

renewRows:cols:

- **renewRows:**(int)*newRows* **cols:**(int)*newCols*

Resets the matrix to be newRows by newCols cells in size. Pre-existing rows and columns will retain their sizes. New columns or rows that have to be added will be the default cell size.

setHeight:ofRow:
- **setHeight:**(NXCoord)*newHeight* **ofRow:**(int)*rowNum*

Resizes the given row to the given height.

See also: ±€**setWidth:ofCol:**

setIntercell:
- **setIntercell:**(const NXSize *)*newSize*

Sets the spacing between the cells.

See also: ±€**setWidth:ofCol:**

setWidth:ofCol:
- **setWidth:**(NXCoord)*newWidth* **ofCol:**(int)*colNum*

Resizes the given column to the given width.

See also: ±€**setHeight:ofRow:**, ±€**totalWidthPerRow**

sizeToCells
- **sizeToCells**

Resizes the Matrix to be exactly big enough for its cells.

sizeToFit
    - **sizeToFit**

Changes the column and row sizes to accommodate the Cell in each with the largest contents.   The width and height of the Matrix frame is then changed to exactly contain the Cells.   Does not redraw the Matrix.

See also: ±€**sizeToCells**

totalWidthPerRow
    - (float)**totalWidthPerRow**

Returns the sum of the widths of all columns.

See also: ±€**heightOfRow:**, ±€**widthOfCol:**

widthOfCol
    - (float)**widthOfCol:**(int)*aCol*

Returns the width of *aCol.*

See also: ±€**heightOfRow:**, ±€**totalWidthPerRow**

write
    - **write:**(NXTypedStream *)*strm*

Writes the object to the typed stream.   Overriden to write the size storage.

See also: ±€**read:**