

---

# Resizing for R3 GUI framework, low-level documentation

Author: Ladislav Mecir

Revision History  
1-Feb-2011/14:41:19+1:00

ALM

## Table of Contents

1. Purpose .....	1
2. Alignment attribute .....	1
3. Datastructures .....	1
3.1. The hgroup/vgroup layouts .....	1
3.2. The hpanel and vpanel layouts .....	4
4. Low level functions .....	5
4.1. REMOVE-FROM-GROUP .....	5
4.2. INSERT-INTO-GROUP .....	5
4.3. CHANGE-LINE-ALIGNMENT .....	6
4.4. RESIZE-GROUP .....	6
4.5. RESIZE-PANEL .....	6

## 1. Purpose

This document describes the implementation details of the R3 GUI resizing replacement subsystem.

## 2. Alignment attribute

At the high level, a graphic object, row, or a set of rows can be aligned vertically using one of the top middle bottom options, or vertically using one of the left center right options.

For example, a graphic element inserted into a row in a group can be aligned to the top of the row, to the middle of the row, or to the bottom of the row. On the other hand, the same element inserted into a column in a layout can be aligned to the left side of the column, to the center of the column, or to the right side of the column.

To be able to use a "common value" that can be used to align the element both in a row as well as in a column, we use the words top-left middle-center and bottom-right.

If a graphic element uses e.g. a top-left alignment, then it is positioned at the top of its row in a group, or at the left side of its column in a layout. Similarly for the other "low level attribute values".

## 3. Datastructures

### 3.1. The hgroup/vgroup layouts

In addition to the data described in the

=url resizing.html

file, the hgroup/vgroup layouts contain the INTERN object having the following attributes:

Variable			Type			Description	
init-pane			pair!			holds the initial pane size, in pixels	
lines			block! containing line objects			Describes how the graphic elements are organized into lines	
minification-index			block! containing integers			Line indices ordered so, that the "hardest to minify" line comes first	

## Line objects

The rows(in a group)/columns(in a layout) block must contain non-empty rows/columns consisting of graphic elements, every row/column is a line object having:

Variable			Type			Description	
	start			integer!		Index of the first element in the line	
	length			integer!		How many elements the line contains	
	init-size			pair!		The initial dimensions of the line, in pixels	
	min-size			pair!		The minimal size the line can	

Resizing for R3 GUI framework,  
low-level documentation

						have, in pixels
	max-size			pair!		The maximal size the line can have, in pixels
	offset			integer!		The offset of the line (only one coordinate makes sense), in pixels
	size			pair!		The size of the line, in pixels
	align			word!		The horizontal alignment of the line, possible values are: left center right
	valign			word!		The vertical alignment of the line, possible values are: top middle bottom
	minification-index			block! containing integer! values		Element indices ordered so, that the "hardest to minify" element comes first
	magnification-index			block! containing integer! values		Element indices ordered so, that the "hardest to

							magnify" element comes first
--	--	--	--	--	--	--	------------------------------------

Note: Depending on the LAYOUT-MODE, only one of the LINE/ALIGN LINE/VALIGN attributes is used.

## 3.2. The hpanel and vpanel layouts

In addition to the data described in the

`=url resizing.html`

file, the hpanel/vpanel layouts contain the INTERN object having the following attributes:

Variable			Type			Description	
init-pane			pair!			holds the initial pane size, in pixels	
init-heights			block! containing integer values			holds the initial heights of layout rows	
min-heights			block! containing integer values			holds the minimal heights of layout rows	
max-heights			block! containing integer values			holds the maximal heights of layout rows	
init-widths			block! containing integer values			holds the initial widths of layout columns	
min-widths			block! containing integer values			holds the minimal widths of layout columns	
max-widths			block! containing integer values			holds the maximal widths of layout columns	

row-minification-index			block! containing integers			Row indices ordered so, that the "hardest to minify" row comes first	
row-magnification-index			block! containing integers			Row indices ordered so, that the "hardest to magnify" row comes first	
column-minification-index			block! containing integers			Column indices ordered so, that the "hardest to minify" column comes first	

## 4. Low level functions

### 4.1. REMOVE-FROM-GROUP

The REMOVE-FROM-GROUP's function purpose is to facilitate subgob removal from a group. In addition to removing the graphic elements it removes also the lines (rows or columns) that become empty. Other affected lines are adjusted to not contain the removed elements.

```
set 'remove-from-group funct [  
    {remove subgob(s) from a group}  
    group [gob!]  
    index [integer!]  
    length [integer!]  
]
```

### 4.2. INSERT-INTO-GROUP

The INSERT-INTO-GROUP's function purpose is to facilitate subgob insertion into a group. In addition to inserting the graphic elements it also adds lines (rows or columns) into the layout. No line is allowed to become empty as a result of such insertion, though.

In addition to inserting graphic objects, it is also possible to insert the word `return`, which causes a line break, i.e. an insertion of a new line, breaking the affected line into two.

```
set 'insert-into-group funct [  
    {insert faces(s) into a group}  
    group [gob!]  
    index [integer!]  
    face [word! object! block!] {RETURN signals line break}  
]
```

## 4.3. CHANGE-LINE-ALIGNMENT

The INSERT-INTO-GROUP function creates new lines using just the default alignment. This function is needed to adjust the alignment of the lines, if different alignment is required.

If the function is called using just a word, the alignment of all lines in the affected group is changed to this value. If a block is used, then the alignments are taken from the block.

If the length of the alignment block is greater than the line count of the affected group, the excess alignments are ignored. If the length of the alignment block is lesser the line count of the affected group, the excess lines remain unaffected.

```
set 'change-line-alignment funct [  
    {changes line alignment for all lines in a group}  
    group [gob!]  
    align [word! block!]  
    valign [word! block!]  
]
```

## 4.4. RESIZE-GROUP

This is the function that actually resizes a vgroup/hgroup (changes the size, computes the new sizes and offsets of graphic elements, and calls their on-resize functions).

```
set 'resize-group funct [  
    {resize a group}  
    group [gob!]  
]
```

## 4.5. RESIZE-PANEL

This is the function that actually resizes a hpanel/vpanel (changes the hpanel/vpanel size, computes the new sizes and offsets of graphic elements, and calls their on-resize functions).

```
set 'resize-panel funct [  
    {resize a panel}  
    panel [gob!]  
]
```