
High level management of MIN-SIZE, MAX-SIZE and INIT-SIZE

Author: Henrik Mikael Kristense

Revision History
Type A
Specification Document

Table of Contents

1. Introduction	1
2. FLEX	2
2.1. Pair Notation	2
2.2. Word Notation	3
2.3. Block notation	4
3. HINT	5

=todos

1. Introduction

The creator of styles are faced with three pairs: MIN-SIZE, MAX-SIZE and INIT-SIZE, each sizes that may need to be defined for a style to work as intended.

A noted side effect is that many small faces contain an identical value in one direction, usually vertically, for all three values. This reveals certain usage patterns that can be shortened into specific actions.

Example of using explicit sizes to create a face that does not resize at all vertically:

```
bar: box [  
  about: "Simple horizontal divider bar."  
  facets: [  
    init-size: 1x2  
    min-size: 1x2  
    max-size: -1x2  
  ]  
]
```

A derivative that tries not to destruct size information from the parent:

```
box: [  
  about: "Simple box"  
  facets: [  
    init-size: 100x100  
    max-size: guie/max-coord  
  ]  
]  
  
bar: box [  
  about: "Simple horizontal divider bar."  
  facets: [  
    init-size: 100x100  
    max-size: guie/max-coord  
  ]  
]
```

```
about: "Simple horizontal divider bar."  
facets: [  
    init-size: 1x2  
    min-size/y: init-size/y  
    max-size/y: init-size/y  
]  
]
```

This can be simplified.

2. FLEX

The idea of FLEX requires two values passed, one for each direction, so here, three methods are presented, one with pairs, one with words and one with blocks.

2.1. Pair Notation

FLEX takes a pair and performs one of two operations of MIN-SIZE and MAX-SIZE:

1. If the value is **-1**, the direction will not resize at all. This happens by setting the MIN-SIZE and MAX-SIZE identical to INIT-SIZE for that direction.
2. If the value is **0**, the direction will not be touched and the existing INIT-SIZE, MIN-SIZE and MAX-SIZE values will not be altered.
3. If the value is **1**, the direction will infinitely resize, by setting the MAX-SIZE to GUIE/MAX-COORD for that direction.

Example, for a box that flexes infinitely and a derived bar that flexes horizontally only:

```
box: [  
    about: "Simple box"  
    facets: [  
        init-size: 100x100  
        flex: 1x1  
    ]  
]
```

A derived bar that flexes horizontally only:

```
bar: box [  
    about: "Simple horizontal divider bar."  
    facets: [  
        init-size: 1x2  
        flex: 1x-1  
    ]  
]
```

Layout examples removes the need to know anything about sizes in order to produce the desired effect and are easier to read:

```
view [box options [max-size: as-pair guie/max-coord 2]] ; the number 2 is necessary know  
view [box options [max-size/x: guie/max-coord/x]] ; this is not possible, currently  
view [box options [flex: 0x-1]] ; no previously stated sizing numbers are necessary to k
```

2.2. Word Notation

Using word notation may be easier to read, but may in this form be too easy to produce illegal char combinations.

Word	X resize	Y resize	Action
none!	None	None	Set MIN-SIZE and MAX-SIZE to INIT-SIZE.
'x	Infinite	None	Set MAX-SIZE/X to GUIE/MAX-COORD/X and set MIN-SIZE/Y and MAX-SIZE/Y to INIT-SIZE/Y.
'y	None	Infinite	Set MAX-SIZE/Y to GUIE/MAX-COORD/Y and set MIN-SIZE/X and MAX-SIZE/X to INIT-SIZE/X.
'xy	Infinite	Infinite	Set MAX-SIZE to GUIE/MAX-COORD.
'h	Restricted	None	Set MIN-SIZE/Y and MAX-SIZE/Y to INIT-SIZE/Y. MIN-SIZE/X, MAX-SIZE/X and INIT-SIZE/X are untouched.
'v	None	Restricted	Set MIN-SIZE/X and MAX-SIZE/X to INIT-SIZE/X. MIN-SIZE/Y, MAX-SIZE/Y and INIT-SIZE/Y are untouched.
'hv	Restricted	Restricted	MIN-SIZE, MAX-SIZE and INIT-SIZE are untouched.
'hy	Restricted	Infinite	Set MAX-SIZE/Y to GUIE/MAX-COORD/Y and everything else untouched.
'xv	Infinite	Restricted	Set MAX-SIZE/X to GUIE/MAX-COORD/X and everything else untouched.

Example, for a box that flexes infinitely:

```
box: [  
  about: "Simple box"  
  facets: [  
    init-size: 100x100  
    flex: 'xy'  
  ]  
]
```

A box that flexes horizontally and is restricted vertically.

```
box: [  
  about: "Simple box"  
  facets: [  
    init-size: 100x100  
    min-size/y: 50  
    max-size/y: 200  
    flex: 'xv'  
  ]  
]
```

A derived bar that flexes horizontally only:

```
bar: box [  
  about: "Simple horizontal divider bar."  
  facets: [  
    init-size: 1x2  
    flex: 'x'  
  ]  
]
```

Layout example:

```
view [box options [flex: 'x']]
```

2.3. Block notation

This performs one action per word and stores words in a block. Combinations are easier to figure out.

The default here is important:

By default, the value could be none! or an empty block, so it would provide no resizing in any direction. Then you would open up resizing, by providing words:

Value	Directional Resize	Action
none!	None	Set MIN-SIZE and MAX-SIZE to INIT-SIZE.
empty block!	None	Set MIN-SIZE and MAX-SIZE to INIT-SIZE.
'x'	Infinite X	Set MAX-SIZE/X to GUIE/MAX-COORD/X.
'y'	Infinite Y	Set MAX-SIZE/Y to GUIE/MAX-COORD/Y.
'h'	Restricted X	Set MIN-SIZE/Y and MAX-SIZE/Y to INIT-SIZE/Y.

'v	Restricted Y	Set MIN-SIZE/X and MAX-SIZE/X to INIT-SIZE/X.
----	--------------	---

This will be easier to derive new values from or to investigate, but might be slower to process and eats more memory?

Example, for a box that flexes infinitely:

```
box: [
  about: "Simple box"
  facets: [
    init-size: 100x100
    flex: [x y]
  ]
]
```

A box that flexes horizontally and is restricted vertically.

```
box: [
  about: "Simple box"
  facets: [
    init-size: 100x100
    min-size/y: 50
    max-size/y: 200
    flex: [x v]
  ]
]
```

A derived bar that flexes horizontally only:

```
bar: box [
  about: "Simple horizontal divider bar."
  facets: [
    init-size: 1x2
    flex: [x]
  ]
]
```

Words that conflict would be overwritten. This case results in an infinitely flexing box:

```
box: [
  about: "Simple box"
  facets: [
    init-size: 100x100
    min-size/y: 50
    max-size/y: 200
    flex: [h x v y]
  ]
]
```

Layout example:

```
view [box options [flex: [x]]]
```

3. HINT

=todo Please fill out any specs regarding HINT management here, if needed.