# The difference between Locations and Dimensions in **grid**

Paul Murrell

November 10, 2026

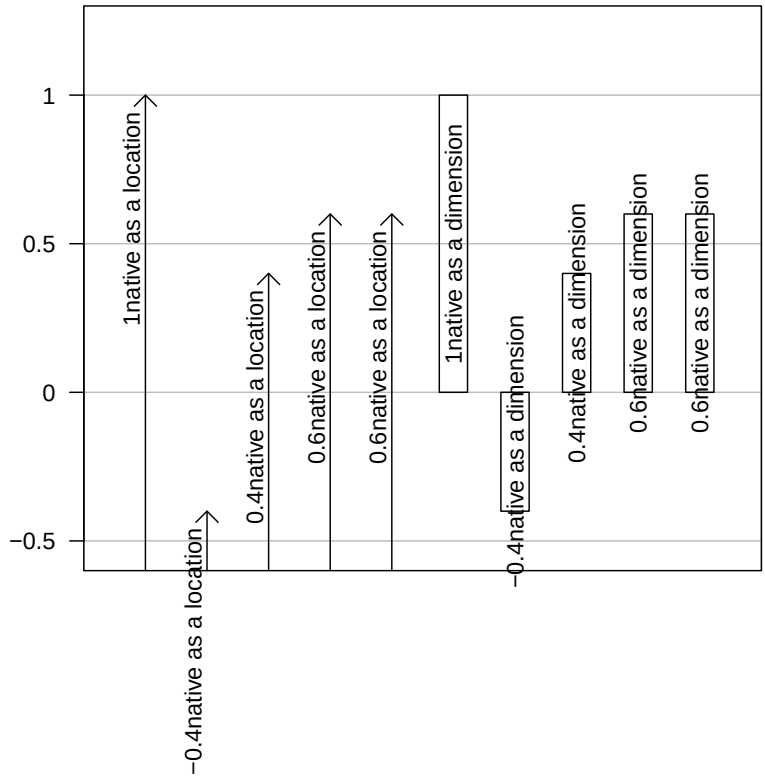
**grid** makes use of unit objects to express both the locations and dimensions of graphical components. For example, the `grid.rect()` function treats its x- and y-arguments as locations within the current viewport, and its width- and height-arguments as dimensions within the current viewport.

These different interpretations of units are usually implicit like this; if its an x- or y-value then its a location, if its a width- or height-value then its a dimension.

The distinction is made at all because, in some coordinate systems, notably `"native"` coordinates, the location $x$ can have a very different meaning from the dimension $x$ – basically, whenever the minimum value of the coordinate system is not zero.

In the specification of simple units, the difference between locations and dimensions is often not noticeable. However, there are a couple of tricky areas:

1. When adding (or performing any arithmetic operation on) units, it is important to keep in mind that locations are added like vectors and dimensions are added like lengths. The following diagram demonstrates the difference:




2. The functions `convertX`, `convertY`, `convertWidth`, `convertHeight` are used to convert from one coordinate system to another. Again, it is important whether the conversion is for a location or for a dimension. The following code demonstrates some results from these functions based on a similar situation to that in the preceding diagram:

```
> pushViewport(viewport(yscale = c(-0.6, 1.3)))
> # Unexpected results?
> convertY(unit(1,'native'), "native")

[1] 1native

> convertY(unit(-.4,'native'), "native")

[1] -0.4native

> convertY(unit(1,'native')+unit(-.4,'native'), "native")

[1] 0.6native

> convertY(unit(1,'native')-unit(.4,'native'), "native")

[1] 0.6native

> # Expected results
> convertHeight(unit(1,'native'), "native")
```

```
[1] 1native
> convertHeight(unit(-.4,'native'), "native")
[1] -0.4native
> convertHeight(unit(1,'native')+unit(-.4,'native'), "native")
[1] 0.6native
> convertHeight(unit(1,'native')-unit(.4,'native'), "native")
[1] 0.6native
> popViewport()
```