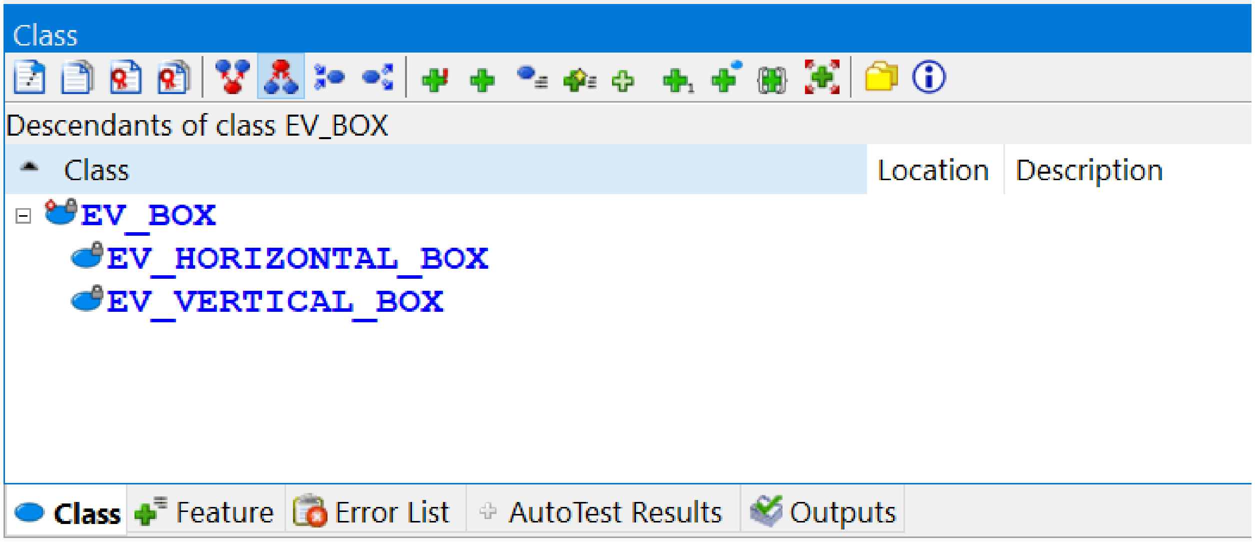
Eiffel Vision2 Demo Project

Containership & Expansion

How GUI containership works in Eiffel Vision2

Containers

There are three basic non-visual containers:



EV_BOX
EV_HORIZONTAL_BOX
EV_VERTICAL_BOX

Horizontal and vertical do what you might think they do. Contents within them are stacked either vertically (top-to-bottom) or horizontally (left-to-right).

You can put vertical boxes in horizontal boxes or reverse them as you like. The nesting of containership is virtually unlimited.

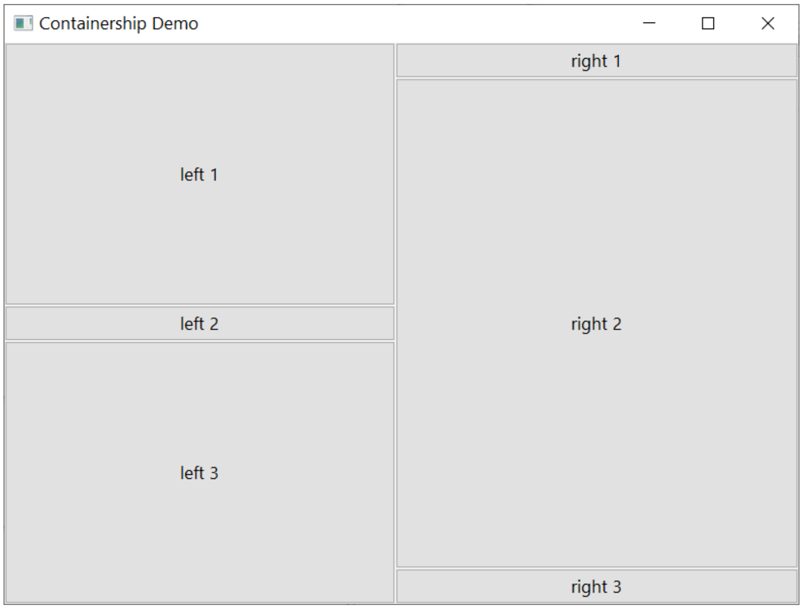
You can also use EV_CELL, which is a container for a single item.

So, the “items” can be either visual GUI element objects or other containers (e.g. vertical, horizontal, and cell).

Item Expansion

Unlike other GUI systems, Eiffel Vision “widgets” (see EV_WIDGET in the Class Context Tool—descendants for an overview of GUI element items, both visual and non-visual) expand to fill their parent containers by default. What does this mean?

When you run the demo-code {CONTAINERSHIP_BOXING_DEMO_WINDOW}, you will see something like:



Notice:

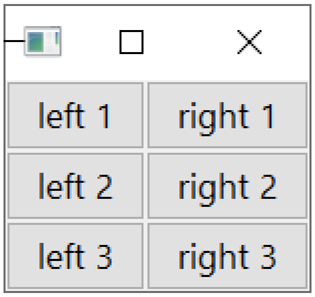
1. The “left_box” and “right_box” have expanded to fill “main_box”, which (in turn) has expanded to fill its containing window, into which it was “extended”.
2. In the left-box, the middle button has been told to “disable_item_expand”, so—it shrinks to just enough height to display its “text”. Otherwise, like left-1 and left-3, it expands left-to-right to fill its parent container.
3. The same is true for the “right_box”, but now it is items 1 and 3 that have had their expansions disabled.

Note that the “expansion” always happens along the axis of the parent in both directions, but disabling expansion only impacts the direction of the parent (e.g. vertical or horizontal depending on the containing parents orientation).

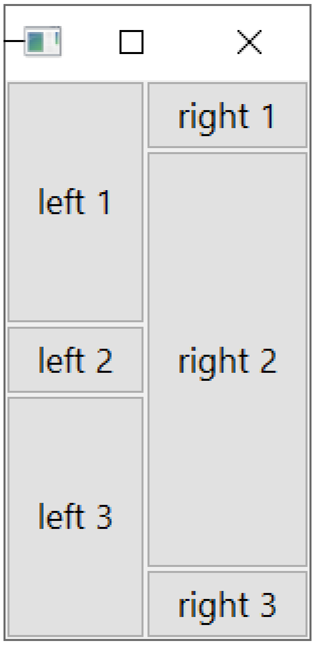
Exercise

Run the program, open the window, and then drag it around to resize it.

Fully collapsed, it looks like:



A little bigger—dragging downward:


right 1
left 1
left 2
right 2
left 3
right 3