Keeps and space-specifiers

by Peter B. West

Table of contents

1 Keeps and space-specifiers in layout galleys....2
1.1 Space-specifiers....2
1.2 Block stacking constraints....2
1.3 Keep relationships between blocks....3

1. Keeps and space-specifiers in layout galleys

The layout galleys (galleys.html) and the layout tree (galleys.html#layout-tree) which is the context of this discussion have been discussed elsewhere. A previous document (keeps.html) discussed data structures which might facilitate the lining of blocks necessary to implement keeps. Here we discuss the similarities between the keep data structures and those required to implement space-specifier resolution.

1.1. Space-specifiers

> **Note:**
> **4.3 Spaces and Conditionality** ... Space-specifiers occurring in sequence may interact with each other. The constraint imposed by a sequence of space-specifiers is computed by calculating for each space-specifier its associated resolved space-specifier in accordance with their conditionality and precedence.

> **Note:**
> 4.2.5 Stacking Constraints ... The intention of the definitions is to identify areas at any level of the tree which have only space between them.

The quotations above are pivotal to understanding the complex discussion of spaces with which they are associated, all of which exists to enable the resolution of adjacent <space>s. It may be helpful to think of *stacking constraints* as *<space>s interaction* or *<space>s stacking interaction*.

1.2. Block stacking constraints

In the discussion of block stacking constraints in Section 4.2.5, the notion of *fence* is introduced. For block stacking constraints, a fence is defined as either a reference-area boundary or a non-zero padding or border specification. Fences, however, do not come into play when determining the constraint between siblings. (See Figure 1.)

Figure 1

block-stacking-constraints.png

> **Note:**
> Figure 1 assumes a block-progression-direction of top to bottom.

In Diagram a), block A has non-zero padding and borders, in addition to non-zero spaces. Note, however, that the space-after of A is adjacent to the space-before of block P, so borders and padding on these siblings have no impact on the interaction of their <space>s. The stacking

constraint A,P is indicated by the red rectangle enclosing the space-after of A and the space-before of P.

In Diagram b), block B is the first block child of P. The stacking constraint A,P is as before; the stacking constraint P,B is the space-before of B, as indicated by the enclosing magenta rectangle. In this case, however, the non-zero border of P prevents the interaction of the A,P and P,B stacking constraints. There is a *fence-before* P. The fence is notional; it has no precise location, as the diagram may lead one to believe.

In Diagram c), because of the zero-width borders and padding on block P, the fence-before P is not present, and the adjacent <space>s of blocks A, P and B are free to interact. In this case, the stacking constraints A,P and P,B are as before, but now there is an additional stacking constraint A,B, represented by the light brown rectangle enclosing the other two stacking constraints.

The other form of fence occurs when the parent block is a reference area. Diagram b) of Figure 2 illustrates this situation. Block C is a reference-area, involving a 180 degree change of block-progression-direction (BPD). In the diagram, the inner edge of block C represents the content rectangle, with its changed BPD. The thicker outer edge represents the outer boundary of the padding, border and spaces of C.

While not every reference-area will change the inline-progression-direction (IPD) and BPD of an area, no attempt is made to discriminate these cases. A reference-area always a fence. The fence comes into play in analogous circumstances to non-zero borders or padding. Space resolution between a reference area and its siblings is not affected.

In the case of Diagram b), these are block stacking constraints B,C and C,A. Within the reference-area, bock stacing constraints C,D and E,C are unaffected. However, the fence prevents block stacking constraints such as B,E or D,A. When there is a change of BPD, as Diagram b) makes visually obvious, it is difficult to imagine which blocks would have such a constraint, and what the ordering of the constraint would be.

Figure 2

block-stacking-keeps.png

1.3. Keep relationships between blocks

As complicated as space-specifiers become when reference-areas are involved, the keep relationships as described in the keeps (keeps.html#Figure1) document, are unchanged. This is also illustrated in Figure 2. Diagram b) shows the relative placement of blocks in the rendered output when a 180 degree change of BPD occurs, with blocks D and E stacking in the reverse direction to blocks B and C. Diagram c) shows what happens when the page is too short to accommodate the last block. D is still laid out, but E is deferred to the next page.

Note that this rendering reality is expressed directly in the area (and layout) tree view. Consequently, any keep relationships expressed as links threading through the layout tree will not need to be modified to account for reference-area boundaries, as is the case with similar space-specifier edge links. E.g., a keep-with-next condition on block B can be resolved along the path of these links (B->C->D) into a direct relationship of B->D, irrespective of the reference-area boundary.

While the same relationships obviously hold when a reference area induces no change of BPD, the situation for BPD changes perpendicular to the parent's BPD may not be so clear. In general, it probably does not make much sense to impose keep conditions across such a boundary, but there seems to be nothing preventing such conditions. They can be dealt with in the same way, i.e., the next leaf block linked in area tree order must be the next laid out. If a keep condition is in place, an attempt must be made to meet it. A number of unusual considerations would apply, e.g. the minimum inline-progression-dimension of the first leaf block within the reference-area as compared to the minimum IPD of subsequent blocks, but *prima facie*, the essential logic of the keeps links remains.