

# Blink paintlist update algorithm details

**Update:** this has landed in <https://codereview.chromium.org/697543002>

Main blink paint list doc is available [here](#).

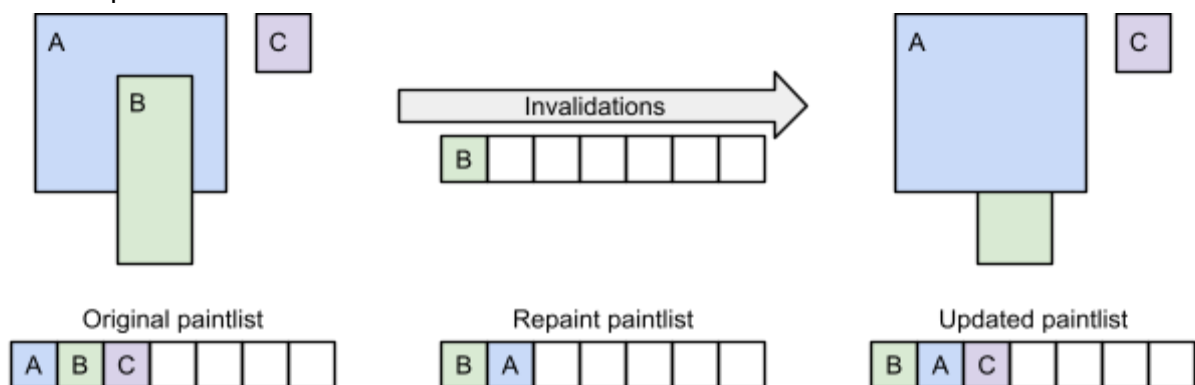
The paintlist update algorithm takes the existing paint list, plus a list of invalidations, and creates an updated paintlist. Creating a paintlist is as simple as running blink's repaint algorithm (which runs the web painting algorithm) and outputs a list of paint chunks. When an invalidation occurs, blink's repaint algorithm produces a paintlist from just the invalidated area and our task is to use this to create the updated paintlist.

Prototype: <https://codereview.chromium.org/604893003>

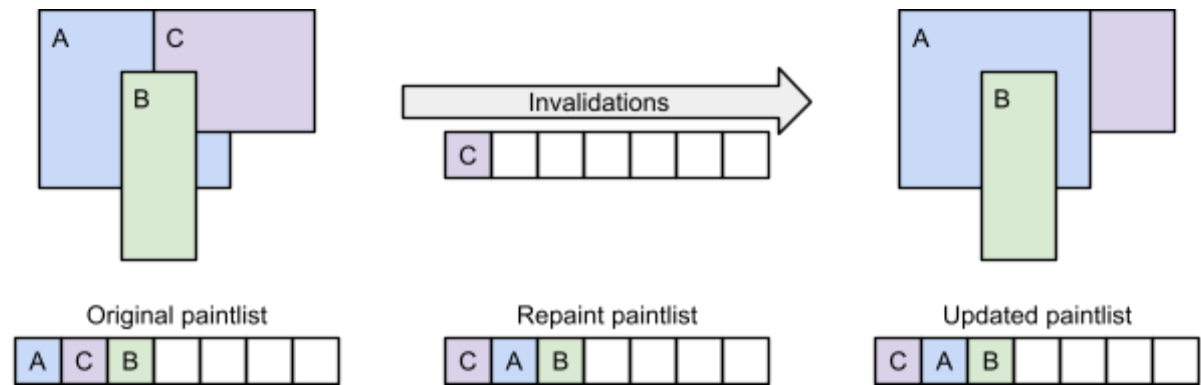
Pseudocode (TODO: make this code):

```
update(original, repaint, invalidations) {  
    original.removeAll(invalidations);  
    Walk both the repaint and original lists in parallel {  
        If the repaint chunk has a corresponding chunk in the original list, consider  
        it a repaint. For repaints, copy over all of the original chunks up to the  
        repaint.  
        Append the repainted chunk to the new paint list.  
    }  
    Copy over any remaining chunks from the original list.  
}
```

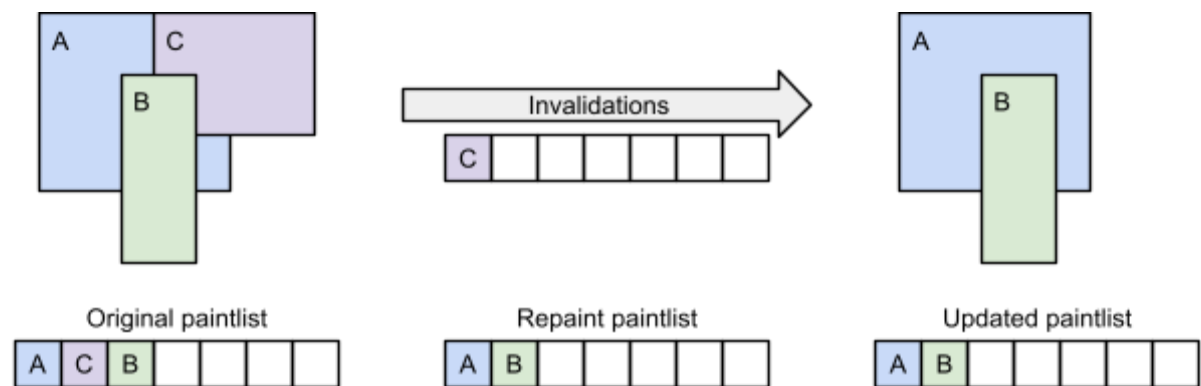
Some examples:



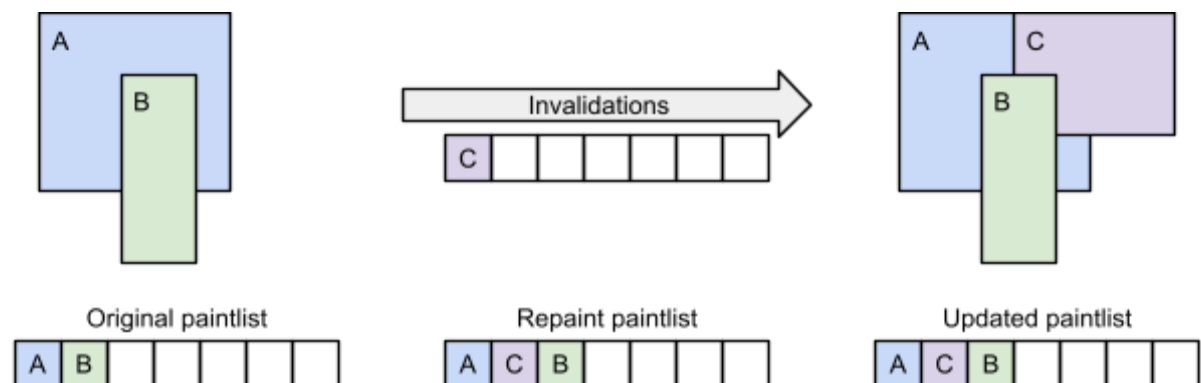
In this example, B is invalidated and moved below A. The update algorithm starts by removing B from the original list. B is then copied from the repaint list to the updated list. A is then recognized as a repaint and is copied to the updated list. Lastly, C is copied over to the updated list.



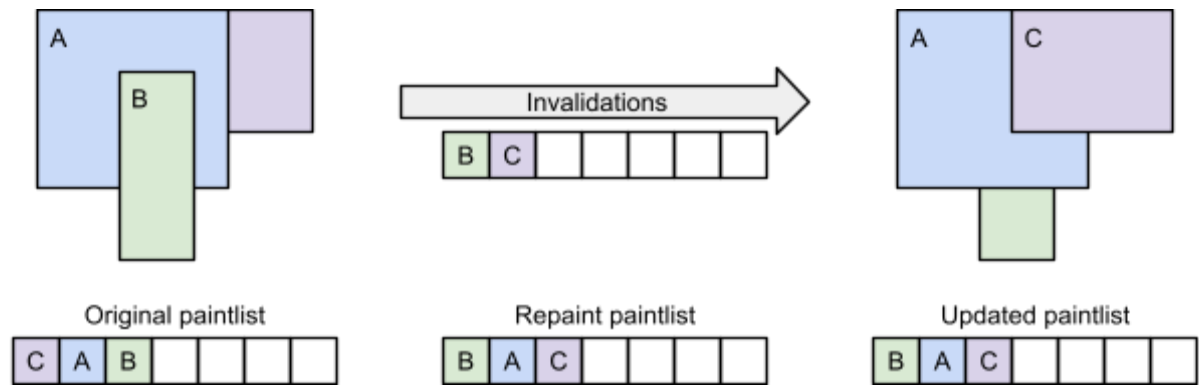
In this example, C is moved to the back.



In this example, C is removed.



In this example, C is added.



In this example, B is moved to the back and C is moved to the front.