Subject: Re: Ctrl: Paint only affected area?
Posted by dolik.rce on Sun, 20 Mar 2011 14:03:25 GMT
View Forum Message <> Reply to Message

Hi Lance

If I am not mistaken, U++ always repaints only the parts that are necessary. The Paint() method always repaints the entire area of the Ctrl, but only parts that need to be updated are actually transfered to the screen (which is the slowest part of the process usually). The Ctrl itself should usually repaint it's whole area. In some cases, you can cache the image inside the Ctrl and change only parts that you know that they need it (e.g. GridCtrl uses this approach I think), but it requires you keep track of what needs to be repainted yourself.

The mechanism works like this: When something needs to be updated, you call its Refresh() method, if only part of Ctrl needs to be updated then use Refresh(Rect r). U++ then calls Paint() on all Ctrls with changed areas and then transfers all the marked areas to the screen. To see which parts of your GUI are repainted and when, call Ctrl::ShowRepaint(50) inside your code, it will help you understand and it is sometimes also quite useful for debugging

There is also a set of methods intended to track the changes inside Ctrl, which can be helpful sometimes: SetModify(), IsModified() and ClearModify(). These are not directly related to the repainting (AFAIK), but they are worth mentioning since they provide common interface that can be often used to optimize the performance (I (mis)use it for caching the Ctrls visuals) while keeping the code readable.

Hope I didn't simplify it too much, but this is actually all I ever needed to know about this

Best regards,
Honza